UTS UNIVERSITY OF TECHNOLOGY SYDNEY

A Framework for SLA Management in SOA-based SDN for Informed Decision Making

by Shuraia Khan

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of
Principle Supervisor: Dr. Farookh Khader Hussain
Co-Supervisor: Dr. Nabin Sharma

University of Technology Sydney
Faculty of Engineering and Information Technology (FEIT)

October 2022

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Shuraia Khan*, declare that this thesis is submitted in fulfilment of the requirements for the award of *Doctor of Philosophy* in the *School of Computer Science, Faculty of Engineering and IT* at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 17/10/2022

Table of Contents

List to Figures

List of Tables

Thesis Summary

Supporting end-to-end Quality of Services (QoS) in existing network architecture is an ongoing issue. Software Defined Networking (SDN), the new network norm, has emerged in response to the limitations of traditional networks. SDN architecture incorporating Service Oriented Architecture (SOA) brings a network notion of a future service-oriented network. Supporting high dependency on intelligent applications is gradually increasing the demand for high-capacity networks with QoS delivery is vital. The absence of ensuring QoS leads to less reliability and a lower level of trust among the interacting parties, which can significantly impact service-based business. In order to address the pressing issues, the existing studies proposed various solutions; however, most of the solutions are application-based and unable to provide a reliable and personalised QoS guarantee in the SDN network. Moreover, no research has been proposed to provide QoS of future Service Oriented SDN. In order to ensure personalised QoS delivery and enhance to preserve the trust relationships, we have proposed Service Level Agreement (SLA) based framework in SOA-based SDN.

The proposed SLA-based framework in SOA-based SDN should address a series of important research questions: (1) How to develop an intelligent method for personalised QoS delivery in SOA-based SDN? (2) How to monitor the QoS of the runtime services in SDN network and pro-actively predict the SLA violation to minimise the risk of SLA violation? (3) How to evaluate the service performance impartially and guarantee a reliable service delivery in SOA-based SDN? (4) How to make an intelligent decision regarding service continuation in considering the deliverable service performance during interactions?

This thesis systematically studies how to effectively solve the aforementioned issues with conceptual, theoretical and experimental guarantees. Due to the significance of the aforementioned issue on SOA-based business, it is essential to determine a process that can preserve the SLA contract and trust relationship between the interacting parties. Specifically, this thesis proposes a novel framework SLA-lifecycle-based model to address the issue mentioned above. The primary feature of the framework is that it enables personalised and

reliable QoS delivery in SOA-based SDN. In addition, the intelligent framework approach can monitor the network in run-time and predict the possible SLA violation; that information also assists in intelligent decision-making regarding the continuation of the SLA. To validate the framework, we have developed a proof of concept (POC) based implementation including various components such as tools and simulation software and expressed the results with discussion in this thesis.

Acknowledgements

I want to acknowledge the grace of God for giving me the knowledge and opportunity to complete my Doctoral Dissertation under the supervision of Professor Farookh Khadeer Hussain and Dr Nabin Sharma.

I wish to acknowledge the efforts of my parents, Md. Jamir Ali Khan and Mrs Nur Rekha, who has always supported me, built courage in me to dream about this degree and work towards it, stayed beside me when I needed and have made uncountable sacrifices for me; I can never thank you enough for all that you have done for me. To my family and brother, Abdullah-Al-Mamun, I can not thank you enough for your unwavering support and sacrifices made for me, and your belief in me was crucial in completing this thesis. To all my family, I hope this thesis is a sign of the good things to come.

Additionally, I am immeasurably grateful to my primary supervisor, Professor Farookh Khadeer Hussain and my Co-supervisor, Dr Nabin Sharma, for their never-ending support, never-ending compassion, encouragement and outstanding guidance. This thesis is as much of their effort as it is mine. I am thankful to Professor Massimo Piccardi and Associate Professor Wenjing Jia, who always believed in me and supported my entire academic and professional career. Moreover, I would like to thank all of the School of Computer Science members for their help, support and guidance during my Higher Degree Research. I thoroughly enjoyed this academic journey and gained invaluable skills to become a better researcher.

It is a pleasure to thank those whose encouragement and support in various stages of this academic journey have made this thesis possible, naming: Dr Zenon Chaczko, Dr Omar Khadeer Hussain, Professor Olivia Fachrunnisa, Professor Guandong Xu, Dr Roksana Huque, Mr Adam Wilson, Mr Curtis Anthony Finn, Dr Niharika Rahman, Dr Ammara Masood, Dr Mohammed Ikram, Professor Touhid Bhuiyan.

Finally, I dedicate this thesis to Professor Farookh Khadeer Hussain, who has been a significant person throughout the work done in this thesis, my parents, my brother and my family.

List to Publications

Journal Articles Published:

1. Shuraia Khan, Farookh Khadeer Hussain, Omar Khadeer Hussain, "*Guaranteeing end-to-end QoS provisioning in SOA based SDN architecture: A survey and Open Issues.*"Future Generation Computer Systemss, Volume 119, 2021, pp.176-187

Referred Conference Articles:

2. Shuraia Khan, Farookh Khadeer Hussain, "*Software-Defined Overlay Network Implementation and Its Use for Interoperable Mission Network in Military Communications.*" Proceedings of the 36th International Conference on Advanced Information Networking and Applications AINA (1) 2022: 554-565.

3. Shuraia Khan, Farookh Khadeer Hussain, *"Evaluation of SLA Negotiation for Personalized SDN Service Delivery."*, Proceeding of the 34th International Conference on Advanced Information Networking and Applications AINA 2020: 579-590.

4. Shuraia Khan, Farookh Khadeer Hussain, *"A SOA based SLA negotiation and formulation architecture for personalized service delivery in SDN."*, Proceeding of the 22nd International Conference on Advances in networked-based Information Systems(NBIS) 2020: 108-119.

5. Shuraia Khan, Adam Wilson, Farookh Khadeer Hussain, *"Contextual information aware optimal communication in radio networks in considering the pervasive computing – A literature review",* Proceeding of the 19th International Conference on Mobile Systems and Pervasive Computing (MobiSPC) 2022, V 203, 127-134.

Chapter 1

INTRODUCTION

1.1 Introduction

This chapter presents an overview of the importance of ensuring Quality of Service (QoS) delivery and preserving trust relationships in service-oriented environments in general and service-oriented software-defined networking (SDN) environments in particular. To realise the importance of personalised and reliable QoS delivery in a service-based SDN business environment, we organise this chapter as follows: Section 1.2 discusses the importance of ensuring QoS in Service-oriented Architecture (SOA) -based SDN and an adjacent relationship between QoS delivery and trust relationship maintenance. Section 1.3 explains the pressing issues related to QoS delivery, service reliability, and preserving trust relationships in service-based SDN environments. The identified pressing issues which lead to the formulation of the research objectives of this thesis are presented in section 1.4. The scope of the thesis is discussed in section 1.5, which clearly outlines what lies within the scope and what lies beyond the scope of this thesis.

Furthermore, the thesis significance is outlined in section 1.6. An introductory briefing to each of the continuing eight chapters of this thesis is given in section 1.7. Finally, we conclude the chapter in section 1.8 and set the scene for the second chapter.

1.2 The Importance of Ensuring Quality of Service in SOA-based SDN

The rapid advancement of computing technology creates a high dependency on internet services, and the demand for various high-performing applications and services with diverse

requirements has increased. Applications such as web surfing, texting, VoIP, email, audio, video conferencing and streaming, online gaming, and e-commerce are required to meet end users' day-to-day demands. These applications and services generate their characteristic flows with different treatments that must be delivered successfully by the internet over a network. The highly diverse and dynamic network demanded by current and emerging applications brings new challenges to service provisioning in future networks (Duan et al., 2016).

SDN introduces itself as an effective solution for current and future technology needs. SDN is defined as a new norm for networks (Karakus & Durresi, 2017) that is capable of supporting the dynamic nature of future network functions and intelligent applications while lowering operating costs through simplified hardware, software, and management (Khan, 2015). The Open Networking Foundation (ONF) defines SDN as follows: This architecture decouples the network control and forwarding functions enabling the network control to become directly programmable and the underlying infrastructure to be abstracted for applications and network services((ONF), 2012)

The logical separation and programmability features simplify the management of the network. For its logically centralised control, dynamically optimised flow and resource management, infrastructure-based abstraction, and large-scale scalability, SDN has attracted the attention of companies, universities, data centres, and network service providers for deployment in their networks to make the best use of it (Karakus & Durresi, 2017).

QoS defines the overall performance of any service. This is a critical component of any service that is being offered. QoS assists the end user in choosing a service with the desired criteria. Moreover, QoS is a critical parameter that assists the decision-making process and builds a good relationship between the service provider and the consumer.

Large-scale networks consist of heterogeneous autonomous systems that need to support diverse applications. These diverse applications generate significant data traffic over networks with different QoS requirements (Duan, 2014). End-to-end QoS provisioning across multiple network domains is a significant challenge and has attracted the attention of network

researchers. The nature of SDN technology can significantly impact future networking by enabling an open, programmable network platform that provides excellent flexibility to support various applications (Duan, 2014). In addition, separating the control and data plane and centralised control ability in SDN offers promising approaches to facilitate inter-domain end-to-end QoS provisioning in future networks (Duan, 2014). Four broad categories of mechanisms have been developed in the existing studies to achieve QoS in SDN architecture: a QoS-based controller design or a new framework-based mechanism, dynamic resource allocation mechanisms, queue or flow management and scheduling-based mechanisms, and QoS-based optimal routing mechanisms.

Moreover, other approaches have been proposed, such as SLA-driven QoS (Gomes et al., 2017; Machado et al., 2014). However, this research is still in its infancy, and there are no concrete implementation results. In addition, none of the approaches can evaluate the QoS guarantee or provide personalised QoS delivery in SDN. Moreover, the existing QoS guaranteed approaches cannot provide a QoS guarantee for the entire SDN network. Most of the approaches are application-based.

1.2.1 SOA Principle in SDN

An argument has been put forward that future internetworking architecture can function with SOA principles, such as the network will be service contact-oriented, loosely coupled, abstracted, reusable, independent, state, discoverable, and composable (Priyanka & Singh, 2014). Cloud computing is a successful development result of SOA. Cloud computing, grid computing, virtualization, and SOA have led the computing industry to a new pathway. Cloud computing offers computing and storage resources in a service-oriented model. As the principle of SOA is to provide a dynamic composition of services, several researchers are approaching some evolutionary architecture in internetworking that aims to achieve the SOA principle by composing several more minor services into specialised services (Martini & Paganelli, 2016). In such a way, the network architecture will be flexible enough to add, change or remove functionality and adapt to specific customer or application needs in changing environmental conditions.

SDN architecture separates the network control and data forwarding functionalities and provides centralised, programmable, advanced network control and highly scalable network services to the consumer (Duan et al., 2016). It allows service provision according to current business needs with unlimited potential in SDN. However, implementing this advancement faces new challenges as SDN's design architecture is insufficient to support these features. To receive a service-oriented network, data plane, and control plane of SDN architecture requires another dimension of decoupling between service functions and network infrastructure to reveal the full potential of SDN (Duan et al., 2016).

SOA offers an effective mechanism to enable flexible interaction among autonomous systems to meet diverse service requirements. Applications of SDN with service-oriented principles are proposed to address the challenging problems of end-to-end QoS provisioning. This principle leads the networking technology to the network-as-a-service (NaaS) paradigm, which makes future carrier networks look more like clouds (Duan, 2014). In this new paradigm, networking resources are abstracted and utilised by users' on-demand in-network services (Duan, 2014). On the other hand, network functions virtualisation (NFV) is a technology that applies cloud networking concepts in the telecommunications space. NFV allows network functions to be deployed as virtualised software instances instead of dedicated hardware appliances. Moreover, NFV enables the creation of logically isolated network partitions over a shared physical network infrastructure, allowing the aggregation of multiple resources as single resources (Papneja, 2022).

To optimise network functionality, IT advancements offer an approach to combining SDN and NFV architecture that makes the network functionalities more service-oriented (Papneja, 2022). Chapter 2 details how several researchers and technology industries are trying to combine SDN and NFV to achieve the SOA principle in SDN. Few studies have implemented simulation or emulation environments; thus, most are still theoretical. Therefore, the successful implementation of this combined architecture is still a challenge. Furthermore, offering QoS in SDN with the service-oriented paradigm is still an open research issue.

On the other hand, maintaining QoS in the network is a long-standing issue for the network industry. With the development of SDNs, revolutionary software-directed network technology brings the new challenge of achieving the required satisfaction level. SDN introduces loose coupling that can provide considerable flexibility in running the entire network based on its interests. This flexibility creates several quality satisfaction levels. Ensuring differentiated satisfaction with the quality of network services introduces new challenges for service providers and creates confusion regarding guaranteed services for network consumers. Controller design, resource allocations, queue scheduling, and optimal routing are four well-known approaches proposed and used by several SDN research scholars. However, none of the existing approaches can provide personalised service delivery in SDN with guaranteed verification.

Moreover, most studies are application-based QoS solutions in SDN, whereas very few studies propose QoS solutions for the entire network and maintaining trust relationships. The following section discusses the pressing issues to guarantee QoS in SOA-based SDN. Moreover, a literature review (Chapter 2) highlights the aforementioned limitations and challenges of the existing approaches in achieving a QoS guarantee in SDN and preserving trust relationships.

This doctoral research emphasises the need for a framework that provides informed decision-making for SLA management in SDN. This framework should provide personalised service delivery to the end user with guaranteed QoS. This QoS would not be applicable for service-oriented-based applications; instead, this framework provides a QoS guarantee for the entire network. More precisely, this thesis proposes a dynamic and reliable SLA-driven framework that will assist the SDN service provider and service consumer in having QoS-guaranteed personalised service delivery in SDN.

The thesis structure is as follows: Chapter 2 reviews the literature in the study area and discusses the comparative analysis and classification of approaches in SDN to achieve a QoS guarantee. Chapter 3 outlines the research challenges and research questions. Chapter 4 presents the proposed research solution model. Chapter 5 discusses the research approach and methodology for service negotiation and requirement formalisation. Chapter 6 discusses the research approach and methodology for service management and proposes a conjoint service

monitoring framework (passive and proactive performance monitoring), and Chapter 7 discusses the research approach and methodology for service evaluation and predicting a possible SLA violation framework. Chapter 8 discusses the research approach and methodology for the service continuity intelligent decision-making framework, and Chapter 9 concludes the thesis and outlines its research contributions and progress.

This chapter outlines and discusses the identified limitations and challenges and formulates the research objectives, scope, and significance. The chapter is organised as follows: Section 1.3 outlines and discusses the pressing research issues; Section 1.4 outlines the research objective; section 1.5 outlines the research scope; 1.6 outlines the research significance; section 1.7 outlines the planning of the thesis and concludes the chapter.

1.3 Issues Related to Quality of Service (QoS), Reliability, and Trust in Service-Oriented Environment

This section discusses related issues regarding QoS, reliability and trust in a service-oriented SDN environment. It should be remarked that the related issues regarding the trust relationship in all SOA domains are outside the scope of this research. In this thesis, we focus on the pressing issues related to ensuring QoS, their inherent relationships, and their direct impact on service reliability and trust relationship. We divide the discussions into two parts. In the first section, we discuss some primary QoS, service reliability, and trust-related issues in commercial service-oriented online systems. Secondly, we discuss the primarily related issues in the research community regarding QoS, service reliability, and trust relationships that need to be addressed. Thirdly, we discuss QoS, service reliability, and trust relationship-related primary issues that need to be addressed in a service-oriented SDN environment.

1.3.1 QoS, Service Reliability, and Trust-based Issues in a Commercial Service-Oriented Online Environment

In the commercial service-oriented online environment, a trust relationship is illustrated in various ways, including reputation rating systems, ranking systems, or recommender systems (Hussain, 2006). A rudimentary version of a ranking system, a rating system, or a combination of these two systems is widely used in commercial systems, e.g. Amazon and eBay (Hussain, 2006).

With the current dependency on online business, many commercial businesses, such as Epinions, BizRate, CNet, MovieLens, Amazon, eBay websites, etc., face trust-related issues regarding their business share. The well-known reputation-based trust system is discussed, whereas eBay is regarded as the most well-known public reputation system based on research identifications (Hussain, 2006). In this system, the eBay member identifies the company's reputation using the user's feedback approach that considers three levels of feedback (negative, neutral, and positive) from the user who had previous interactions with eBay (Hussain, 2006). A formal definition of the concept of reputation and the rating range details are discussed in Chapter 6. Therefore, the following pressing issues regarding trust relationships in the current online-based systems are identified.

1. An increasing number of fraud complaints decrease the trust and reliability of online-based business systems. For example, of the complaints received by the Federal Trade Commission in 2021, 2.8 million fraud reports related to consumer fraud complaints related to customer fraud complaints(Federal Trade Commission 2022). In addition, the Federal Trade Commission data shows that consumers reported losing more than $5.8 billion to fraud in 2021, an increase of more than 70 % over the previous year(Federal Trade Commission 2022).

2. The currently used feedback measures are insufficient (e.g. eBay's feedback measures) in capturing and expressing the seller's satisfaction in terms of the buyer's behaviour in their interaction and vice versa (Hussain, 2006).

3. The static nature of reputation is not adequate to capture a true reflection of the behaviour of a particular business that is subsequently manifesting the changes in the user's behaviour in a transaction carried out over time (Hussain, 2006).

4. A single reputation score for a given user is inadequate to reflect, manifest and capture the different capabilities that a particular user may have in a different context.

5. Current service-based businesses do not adopt any means for making direct trust value-based decisions regarding interactions. The current dynamic nature of reputation modelling is employed in an ad-hoc manner and is not based on any mathematical theory or foundation.

1.3.2 The Pressing Issues of Research on QoS and Trust Maintenance in a Service-Oriented Environment

Given the importance of service reliability and trust maintenance, the research community has given much attention to this issue. As a result, significant progress is demonstrated in several directions. An unstructured open community environment is similar to the service-oriented environment(SOE), where the current trend presents a movement from direct interactions involving a small closed community to static binding toward open indirect, and dynamic interaction (Chang et al., 2006). In such an environment, trust is crucial in anonymous, remote, heterogeneous SOEs and business communications. Trust is defined as the belief the service consumer has in the service provider's willingness and capability to deliver mutually agreed services in a given context and in a given time slot. Hence, in order to maintain long-term business interactions in a service-based virtual environment, trust relationship maintenance needs to be given the highest priority.

On the other hand, QoS in a service-oriented network environment (e.g. the cloud, service-oriented SDN) is defined as the fulfilment of the service agreement or mutually agreed-on network services that need to be delivered with a satisfaction guarantee (Chang et al., 2006). In an SOE, trust and QoS are significant concerns. On the other hand, introducing trust and trustworthiness technology, including recommendation and rating systems, provides some degree of control and QoS.

The research by (Chang et al., 2006) finds a strong co-relation between QoS rating and trustworthiness measurement systems and demonstrates that the trustworthiness measurement system intends to present the QoS or the quality of the business service providers to the open networked environment and to help end users choose the most appropriate service or business provider. Moreover, QoS rating systems assist business build their reputations, values, and consumer confidence (Chang et al., 2006). In the service-oriented environment, in the context of service providers, the trustworthiness concept often refers to the QoS that a business organisation provides. For example, we could say that the service from Insitec Ltd. is excellent instead of saying that the service from Insitec Ltd. is trustworthy.

1.3.3 The Pressing Issues of Research on Guaranteeing Quality of Services in SOA-based SDN

In order to support the building and maintenance of a trusting relationship and trusted environment in a service-oriented SDN, a reliable, recognized, and common framework assists in ensuring, maintaining, and delivering QoS in service-based business interactions. Moreover, such a methodology assists in guiding the behaviour of both service provider and consumer to sustain the agreed contract (SLA) and trusting relationship between both parties. The existing research provides numerous discussions on ensuring QoS in an SOA environment and SDN environment that assists in delivering reliable services and preserving trust relationships between both parties. In addition, the following research issues emerged from the literature review:

1. There is no reputation-based SLA approach to guarantee QoS delivery in SDNs. This reputation-based SLA approach can be used to select a suitable SDN service provider based on consumer requirements.

2. No existing study provides personalised SLA and receives QoS based on their selection. Personalisation is an advanced feature of service selection which means rather than receiving a whole service pack, the consumer can customise the service request based on their needs.

3. There is no service satisfaction measurement /evaluation technique that can verify the level of service which is guaranteed in the SLA.

4. There is no or very little research that can provide an end-to-end QoS guarantee of the whole SDN network. Most of the research is able to provide an application-based QoS only.

5. Very little research has been conducted on proposing SDN in SOA-based environments. However, this research is still in an immature stage for implementation in a production environment.

6. There is no research that provides QoS-guaranteed and reliable SDN services in SOA-based environments.

From the aforementioned identified research issues, several research challenges arise, as described in the following. This thesis aims to address the following challenges with proof of concept.

Lack of offers for personalised service delivery in SDN.

The term "personalisation" is defined as meeting the customer's needs more effectively and efficiently where the "one size fits all" idea is not applicable. Personalisation is achieved through customer data and predictive technology, which increases customer satisfaction. The term customisation has a slightly different meaning from personalisation. Customisation is a method of changing something in order to fit the needs or requirements, and this can be achieved when a user manually makes changes to achieve his preferred experience. By offering personalisation, companies can empower customers to make their services feel like their own. Effective personalisation starts with good quality data, which means collecting all the necessary customer information and preparing the product or services upon customer demand. In the service-oriented computing industry, personalisation is defined as an advanced feature of service selection which means rather than receiving a whole service pack, and the consumer can customise the service request based on their needs. In the service contract stage, the

consumer receives the services on a pay-as-you-go basis. Unfortunately, no existing study can personalise their network services in SDN. As a result, there is no opportunity to receive QoS based on their selection. Moreover, the network service consumer has to pay for all the offered services which they might not need.

Offering a reliable approach to provide guaranteed QoS delivery in software-defined networks.

Given the flexibility of SDN, several SDN providers offer a range of network services using the term "QoS guarantee". Several scholars in their case studies use the term "guaranteed QoS"; however, all researchers explain QoS guarantee in their own way. The increase in network service offerings and the adoption of a dimension of service with its own appealing QoS guarantee mechanism makes SDN service provider comparison a time–consuming and complicated process. However, no existing approaches can be used to select a suitable SDN service provider by considering their previous reputation history. Moreover, there is no or very little research that is able to provide an end-to-end QoS guarantee of the whole SDN network. Most of the existing QoS-guaranteed techniques primarily provide application-based solutions.

Lack of measurement for the service satisfaction guarantee level of SDN.

The term satisfaction has been defined as "Whatever a reasonable person would expect from a product or service" (DR, 2013). A satisfaction guarantee is usually used in a service contract in which the service provider defers to the buyer's unilateral discretion as to whether the goods or services offered are acceptable. Several SDN service providers offer their SDN services with a QoS guarantee. However, the term "guarantee" is very ambiguous, and as demonstrated in the literature review, achieving a QoS guarantee in one research study may differ significantly from another. Moreover, most research investigates application-specific QoS rather than customer expectation-specific QoS. In addition, no such approach can compare customer expectations defined in a contract and the services the customer received. As a result, there is always a deficiency in measuring a service satisfaction guarantee in SDN. This gap adversely

impacts the management of QoS as consumers' expectations help improve the overall service quality and lead to a service satisfaction guarantee.

Lack of QoS guaranteed SDN in SOA environments.

SOA is a computing concept where services can be contact-oriented, loosely coupled, abstracted, reusable, independent, and composed. As the purpose of SOA is to provide a dynamic composition of services, several researchers are developing evolutionary architecture in internetworking that aims to achieve the SOA principle by composing several more minor services into specialised services (Martini & Paganelli, 2016). The advantages of combining the SOA principle in SDN architecture are massive and are considered a feasible thought by the researcher. Some research scholars propose an approach to implementing SDN in the SOA concept. However, this research is still in a very immature stage as most of the research is conceptual. From the above challenges, it is evident that SDN implementation in an SOA-based environment is still in the conceptual stage. Currently, there is no concrete model to define a stable SDN in an SOA environment. Thus, the behaviour of implementing QoS approaches in an SOA-based SDN is very unpredictable.

1.4 Objectives of the Thesis

The previous sections outline the role and the importance of maintaining QoS and trust relationships once trust has been determined and established in enabling trust-based business transactions such as service-oriented business environments. In addition, Section 1.3 highlights some of the pressing issues related to personalised QoS management and trust relationships that need to be addressed in the service-oriented SDN environment. This thesis proposes solutions to some of the trust issues (Section 1.3) in service-oriented SDN environments by proposing a methodology to semantically and quantitatively model the QoS and trust behaviour and demonstrating a co-relation between them that an end user/service requestor has in a product/service provider respectively. The objectives of this thesis are summarised as follows:

Objective 1: To propose a definition of trust maintenance/preservation, trust-based interaction or relationships, time-space of interaction or relationship, third-party agent, online monitoring, proactive continuous monitoring, passive performance monitoring, service-oriented environment (SOE), quality of service (QoS), reputation-based trust decision or reputation-based trust modelling or reputation-based decision making.

Objective 2: To develop an intelligent framework to personalise service delivery in SDN.

Objective 3: To develop an intelligent framework to monitor SDN services and provide predictive decisions of proactive service violations.

Objective 4: To develop a model to measure the QoS guarantee level in terms of the reliability of services.

Objective 5: To develop an intelligent decision-making framework for service continuity.

Objective 6: To develop an experimental prototype to validate the methodological frameworks.

Hence, this research does not emphasise any approach to determining trust value. However, this research emphasises a) how both interacting parties (service consumer and service provider) can preserve or even progress the existing level of the trust relationship beyond the minimum threshold; and b) how to perform a successful transaction in a service-oriented SDN environment from the trust-building to trust maintenance and trust re-adjustment stage.

1.5 Scope of the Thesis

This thesis's methodology will enable personalised QoS delivery in service-oriented SDN. In addition, through the developed methodology, this thesis enables the service consumer and service provider to build a positive trust relationship and maintain their trust level after trust has been determined and established. Moreover, this research enables a service consumer and service provider to make a trust-based decision as to whether to interact with a given provider/consumer entity by taking into account both the context and the time at which the trusting agent intends to carry out the interaction.

Furthermore, this thesis proposes and verifies a methodology by which the service consumer can determine whether the service provider will perform according to the mutually agreed approach. Similarly, the service provider can also determine whether the consumer will respond according to the mutually agreed behaviour. Therefore, the scope of this thesis is as follows:

1. This thesis employs the activities and behaviour of offer-related trust (Fachrunnisa, 2011). Fachrunnisa (2011) argues that the following two areas of trust can influence the relationship between the service consumer and the provider: a) *person-related* trust influence (e.g. sympathy or empathy, graciousness, and consumer/service representative similarity); and b) *offer-related* trust influence (e.g. customisation, competence, reliability, and timeliness).

   This thesis aims to guide service-oriented business relationships where all the transactions are service-based; therefore, *person-related trust* does not meet the primary focus of the thesis and, thus, stays outside of the scope of this thesis.

2. Hence, this thesis deals with *offer-relate trust* only. Regarding the longevity of the trust relationship, (Fachrunnisa, 2011) states that as the length of the relationship grows, the characteristical effect of *offer-related* trust becomes more significant.

3. This thesis focuses more on the perception of soft trust than *hard trust*. Hard trust represents information derived from security mechanisms such as identity keys, credentials, and certificates, while *soft trust* contains information that is implied from the experience and observation of others(Fachrunnisa, 2011). Therefore, identity trust stays outside of the scope of this thesis.

4. This thesis focuses on maintaining trust in service-oriented SDN environments. Other environments, such as infrastructure-based settings, stay outside the scope of the thesis.

5. This thesis is focused on the interactions that are interesting for medium-to-long-term relationships. A contractual agreement borders the interaction between a service

consumer and the service provider. Hence, short-term and non-contractual interaction-based transaction stays out of scope in this thesis.

1.6 Significance of the Thesis

To the best of our knowledge, this thesis is the first and only attempt to semantically and quantitatively model the SLA-lifecycle-based approach to achieve personalised QoS delivery in service-oriented SDN. This thesis attempts to work up the trust-building, maintenance and decline stage referred from the trust evaluation model. Therefore, the significance of this thesis arises from the following:

1. This thesis defines the perceptions of the elements of the trust evolution model (trust-building, trust maintenance, and trust decline) along with the specific characteristics and features of trust and relationships in each stage of the trust evolution inspired by Fachrunnisa's thesis (Fachrunnisa, 2011). Moreover, the thesis defines the concepts of trust, service trust, trust relationship, reputation, and service reputation from the perspective of SOA. To the best of our knowledge, the distinction between these concepts for service-oriented SDN environments has not previously been characterised and presented.

2. This thesis evolved into an SLA lifecycle-oriented methodology intending to achieve personalised QoS delivery and informed decision-making in service-oriented SDN. By proposing this methodology, we aim to build and maintain a trusting relationship between the service consumer and the service provider; therefore, the significance of this thesis also arises from the following:

    a. This thesis proposes a framework for the service negotiation and formulation of service requirements. This framework will guide the service consumer and the service provider in constructing an SLA for their forthcoming interaction. To our knowledge, no literature considers including pre-activities such as service

requirements formulation before negotiating the service requirements in the service-oriented SDN domain. Moreover, as the primary interaction carried out during the trust maintenance stage, the service requirements being formulated and negotiated can be derived from several previous transactions and will operate in the trust-building stage. Furthermore, the existing literature does not discuss how to construct an SLA for forthcoming business interactions in order to sustain trust relationship maintenance when the SDN operates on a service-based platform.

b. This thesis proposes a framework to manage the services using passive and proactive continuous runtime network performance monitoring. With this framework, the service consumer and the service provider can keep track of their deliverable service performance during their interactions in a runtime environment. Thus, the framework assists in having a transparent realization in terms of deliverable services that are agreed upon in order to maintain the trust relationship. To our knowledge, a conjoint approach-based framework for managing the services has not been proposed in the existing literature. The literature is inclined to be more involved with conducting a performance evaluation at the end of the interaction. There is no concern about closely monitoring the performance during the interaction. Proactive continuous monitoring provides a platform for determining discrepancies between the agreed and actual performance in time before it reaches unmanageable levels. On the other hand, passive performance monitoring analyses runtime traffic and prioritises the traffic according to the service need.

c. This thesis proposes a framework for predicting SLA violations by evaluating the service performed during the interaction. We designed a mechanism to evaluate the deliverable services through the collected dataset using proactive continuous performance monitoring and passive performance monitoring. We also introduce an alert system that assists in predicting possible SLA violations and notifying the service evaluation agent. The existing literature reviews do not suggest any approach to predicting SLA violations, and there is no opportunity for the service provider to intervene early before the violation occurs. To our knowledge, no framework has been proposed to predict SLA violations by implementing the early

alert system that supports the service maintenance stage by successfully delivering services without interruptions.

d. This thesis proposes a framework for service continuity decision-making. The service continuity decision-making is, unlike in the existing literature, conducted during the trust maintenance stage. This re-adjustment considers the value of the service performance, which is an intermediate performance assessment because of proactive continuous monitoring. To our knowledge, such service continuity decision-making, intended explicitly for activities in the trust re-adjustment stage, has not been discussed in the literature.

3. This thesis validates the proposed methodology by conducting several experimental simulations to build and maintain trust relationships in service-oriented SDN environments.

1.7 Thesis Plan

This thesis provides a complete methodology for SLA-based QoS guaranteed personalised service delivery in service-oriented SDN. In order to achieve the aforementioned research objectives, this thesis is divided into nine chapters. This section provides the following high-level overview of each chapter:

Chapter 2: Chapter 2 presents an extensive review of the existing approaches of QoS in SDN architecture in a service-oriented environment. It broadly divides the relevant studies into five categories: QoS-based controller design, resource allocation-based approach, queue scheduling and management-based approach, QoS-driven optimal routing, and SLA-based quality management in SDN.In addition, we compare the working of these techniques against the identified requirements of guaranteeing end-to-end QoS provisioning in SOA-based SDN

architecture and present directions for future research. Moreover, problems stemming from the current literature concerning QoS in SOA-based SDN architecture are identified in this chapter. Additionally, the comprehensive survey-based literature review identifies the research issues we address in this thesis. This chapter illustrates that the research issues we address in this thesis have not been previously addressed in the literature.

Chapter 3: This chapter formally explains the research issues we address in this thesis. We suggest a set of definitions of the terminologies used while defining the problem. In addition, we break the research issue into six cohesive research issues in this chapter. Therefore, we define each of these research issues formally. Subsequently, we outline the solution proposal and select research methods to solve the identified research issues.

Chapter 4: This chapter presents an overview of the solution to the six research issues identified in Chapter 3. Chapter 4 also provides a high-level overview of each chapter containing detailed solutions for the identified research issues. In addition, this chapter shows the conceptual descriptions of the trust evolution model and stages of the trust evolution model (building trust, maintaining trust, and declining trust), and each stage is associated with trust and relationship characteristics.

Chapter 5: Chapter 5 presents the framework for formalising and negotiating service requirements and enables personalised service delivery in SOA-based SDN. We argue that the negotiation framework can be the foundation of a trust-building relationship between the service consumer and provider. Therefore, we identified three requirements to achieve personalised service delivery in SOA-based SDN. The proposed framework enables the service provider and consumer to articulate their service requirements, formulate them using a structured mechanism, and determine the prioritisation of service criteria. This chapter also presents a framework that both service consumers and service providers can use to negotiate and address any conflicts.

Moreover, we illustrate the whole process of how the service requirements being negotiated can translate into the construction of an SLA using a case study. Furthermore, the framework

proposes an approach to assist in making an informed selection from the perspective of the service consumer and the service provider. The framework consists of five stages to formulate an SLA, and details of the framework are demonstrated in this chapter.

Chapter 6: A service management framework comprising proactive and passive continuous performance monitoring to deliver QoS in SOA-based SDN is presented in Chapter 6. The proposed framework allows a third-party agent to execute proactive and passive performance monitoring between a service consumer and a service provider during the business transaction. In this framework, we propose that passive and proactive monitoring are both critical in their way. Passive monitoring represents the actual performance with a statistical analysis of the real-time data. At the same time, active monitoring generates predictive results to introduce an early intervention for potential network issues and violations of the agreement and to maintain visibility. We developed the proof of concept-based implementation and demonstration and analysed the results, followed by the discussions presented in this chapter.

Chapter 7: Chapter 7 presents a service evaluation and violation prediction framework for impartial and trustful service delivery in an SOA-based SDN. The framework is implemented during the interaction by a third party or an intermediate agent that the interacting parties agreed upon. Thus, the proposed framework facilitates the maintenance of trust using service performance records. The fundamental hypothesis of our SLA-based framework is that a trust relationship has already been established during interaction and is required to sustain the trust relationship by consistently monitoring, examining, and evaluating the interaction. This chapter presents our service evaluation and violation predicting framework for preserving SLA and achieving service reliability. The implementation and results are described in this chapter, followed by discussions.

Chapter 8: Chapter 8 presents an impartial and trustful service continuity decision-making framework for service delivery in SOA-based SDN. A third party carries out the framework in the presence of the consumer party during the interaction, especially at the midpoint of the interaction that the interacting parties agreed upon. We have mentioned in the previous chapter that the fundamental hypothesis of our SLA-based framework is that a trust relationship has already been established during an interaction, and it is necessary to sustain the trust relationship by consistently monitoring, examining, and evaluating the interaction. Hence, both

parties need to have a transparent picture of the deliverable services by evaluating the deliverable services and identifying any SLA violation during the interaction. Moreover, we know that any violation involves penalties, including financial and reputation. Hence, both parties need a framework that performs an impartial assessment of the entire interaction and assists in service continuation decision-making. The framework implementation and results are discussed in this chapter, followed by discussions.

Chapter 9: Chapter 9 concludes the thesis by defining the problems that are addressed in this thesis. A detailed discussion of the contributions of this thesis in terms of the existing body of literature is presented in this chapter. Moreover, a summary of the results of our work, along with the potential directions for future work that are identified in this research, is presented in this chapter.

1.8 Conclusion

In this chapter, we introduced the importance of ensuring QoS in a service-oriented SDN environment and the importance of ensuring QoS in order to maintain trust relationships once trust has been determined and established. We then presented several research issues related to QoS, service reliability and trust-based issues in a commercial service-oriented online environment and the pressing issue of research into guaranteeing QoS in an SOA-based SDN environment. In particular, this chapter indicates that to maintain a trust relationship, there is a fundamental need to make an effort on it; however, to the best of our knowledge, in the literature, no complete methodology has been proposed to assist in trust building, either in the maintaining, declining or re-adjusting stages. The existing research focuses mainly on delivering QoS in SDN and service-oriented environments. Regarding trust modelling and management, the literature shows minimal concern in determining the value of the trust or making a trust decision regarding the trusted agent.

Additionally, we discuss the identified challenges that this thesis aims to address. In addition, we stated and discussed the objectives of this study, followed by defining the scope and

significance of this thesis in enabling a service provider to maintain the QoS and a trust relationship with the service consumer. Finally, the plan of this thesis was presented.

In the next chapter, we present an overview of the existing literature on achieving QoS in SOA, SDN and SOA-based SDN and determine the research issues that have not been addressed. The objective is to ensure that the research issues we address in this thesis have not been addressed previously.

1.9 References

(ONF), O. N. F. (2012). *ONF Software-Defined Networking (SDN) Definition*. Retrieved 23/01/2022, from https://opennetworking.org/sdn-definition/

Chang, E., Hussain, F., & Dillon, T. (2006). *Trust and reputation for service-oriented environments: technologies for building business intelligence and consumer confidence*. John Wiley & Sons.

DR, B. B. (2013, 11th May). Satisfaction has a legal meaning. *The Customer Institute*.

Duan, Q. (2014). Network-as-a-service in software-defined networks for end-to-end QoS provisioning. 2014 23rd Wireless and Optical Communication Conference (WOCC).

Duan, Q., Ansari, N., & Toy, M. (2016). Software-defined network virtualization: an architectural framework for integrating SDN and NFV for service provisioning in future networks. *IEEE Network, 30*(5), 10-16.

Fachrunnisa, O. (2011). *A methodology for maintaining trust in virtual environments*, Curtin University].

Federal Trade Commission (2022). *New Data Shows FTC Received 2.8 Million Fraud Reports from Consumers in 2021, Reported fraud losses increase more than 70 percent over 2020 to more than $5.8 billion*. https://www.ftc.gov/news-events/news/press-releases/2022/02/new-data-shows-ftc-received-28-million-fraud-reports-consumers-2021-0

Gomes, R. L., Madeira, E. R., & Bittencourt, L. F. (2017). Mechanisms for management of SLA for virtual software defined networks based on QoS classes. 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM).

Hussain, F. K. (2006). *A Methodology for Trust Management in Service Orientated*

Environment, Curtin University of Technology].

Karakus, M., & Durresi, A. (2017). Quality of service (qos) in software defined networking (sdn): A survey. *Journal of Network and Computer Applications, 80*, 200-218.

Khan, S. (2015). *Software Defined Network for Energy Efficient Data Centre Operations* [Masters Report].

Machado, C. C., Granville, L. Z., Schaeffer-Filho, A., & Wickboldt, J. A. (2014). Towards SLA policy refinement for QoS management in software-defined networking. Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on.

Martini, B., & Paganelli, F. (2016). A service-oriented approach for dynamic chaining of virtual network functions over multi-provider software-defined networks. *Future Internet, 8*(2), 24.

Papneja, R. (2022, 24/09/2018). *5G brings a world of new opportunities to service providers. However, these benefits are not realized if networks are not up to the task at hand*. Prodapt Chase Extraordinary. Retrieved 14/04/2022, from https://www.prodaptconsulting.nl/wp-content/uploads/2012/11/Prodapt-Consulting-White-paper-SDN-NFV-2015-V1_1-EN.pdf

Priyanka, M., & Singh, K. J. (2014). An Idea for Improvising the Efficiency of SDN Based Business Design with SOA. *Int. J. Eng. Sci. Innovat. Technol.(IJESIT), Joirnal Articla, 3*(3).

Chapter 2

LITERATURE REVIEW

2.1 Introduction

Ensuring end-to-end quality of service (QoS) in traditional network architecture is an issue that needs to be resolved. Software-defined networking (SDN), the new network norm, was proposed to address the limitations of a conventional network. Some benefits of SDN are its ability to provide a global networking view, its programmability, its decoupling of data, and its control plane. Integrating SDN architecture with SOA brings a new network notion for the future service-oriented network. Researchers from academia and industry have proposed and developed several solutions to address the current QoS limitations in SDNs. This review paper aims to examine the effectiveness of those techniques in SOA-based SDNs.

We divide the relevant studies into five categories: QoS-based controller design, resource allocation-based approach, queue scheduling and management-based approach, QoS-driven optimal routing, and service-level agreement (SLA)-based quality management in SDN. We compare the working of these techniques against the identified requirements of guaranteeing end-to-end QoS provisioning in SOA-based SDN architecture and present the directions for future research.

We oriented this chapter starting with a brief background of the literature review research in section 2.2. Then we have identified and determined some fundamental requirements to guarantee end-to-end QoS provisioning in SOA-based SDN is discussed in detail in section 2.3. As mentioned above, we divide the relevant studies into five categories, and we have discussed all of the five categories explicitly in sections 2.4, 2.5, 2.6, and 2.7. Moreover, we have identified some existing studies in SLA-based quality management and included them in section 2.8. The gap that we have identified after reviewing the existing related literature and the discussions are presented in section 2.9. we conclude the chapter in section 2.10.

2.2 The background of the literature review research

The success of rapidly advancing computing technologies and applications, such as web browsing, email, VoIP, audio and video conferencing, streaming, online gaming, texting, and e-commerce, is highly dependent on the internet. The importance of need and humankind's dependence on these technologies was evident during the COVID-19 pandemic. Each of these technologies has its characteristic flows and thus demands a highly diverse and dynamic network to deliver on their requirements. From the viewpoint of service provisioning, this brings different challenges, ranging from ensuring the security of communication to guaranteeing the QoS being delivered (Duan et al., 2016). This paper focuses on providing the QoS aspect during service provisioning. Thus, we focus on the capability of the network to adapt to the need of the technology and application to deliver the expected service. Traditional networks are decentralized in nature and utilize a static architecture. This places numerous constraints on it to adapt dynamically to the requirements of the application. SDN as technology was proposed to address these limitations of traditional networks. SDN is a model that can support the dynamic nature of intelligent applications while reducing operating costs through simplified hardware, software, and management (Karakus & Durresi, 2017) (Khan, 2015). Unlike a traditional network, SDN decouples the data and control planes and centralizes the controlling process of the whole network (Karakus & Durresi, 2017). This enables the centralized control of the network, dynamic management, and optimization of the flows, resources, and infrastructure. As a result of these advantages, SDN has attracted significant attention from network service providers to make the best use of their networks (Karakus & Durresi, 2017). Researchers have also explored different ways by which SDN can be applied, one of these being SOA-based SDN.

As the principle of SOA is to offer a dynamic composition of services, SOA-based SDN is a network that is service-based, loosely coupled, reusable, and can be abstracted (M. Priyanka, 2014). In other words, SOA-based SDN achieves the SOA principle in network management by incorporating several more minor services into specialized services (Martini & Paganelli, 2016). This makes the network architecture flexible and adaptable to scale up or down

according to specific application needs in changing environmental conditions. Such an architecture has various advantages, especially in autonomous systems where the required network capacity can meet their diverse requirements at different periods. However, benefits can be achieved; other challenges need to be addressed to guarantee end-to-end QoS provisioning in SOA-based SDN architecture. Some of these challenges are (a) negotiating the network required as a service to achieve the expectations, (b) selecting a capable network provider that can deliver on the negotiated requirements, and (c) ensuring that the network being delivered meets the QoS requirements by managing and making proactive decisions that will ensure the expected expectations are met. As shown in Figure 1, these challenges as tasks are not present in managing traditional SDNs. However, from the viewpoint of SOA-based SDN, they are essential to address the networking resources to be abstracted and utilized on-demand by users or autonomous systems in the form of a network-as-a-service (NaaS) paradigm(Duan, 2014). To achieve this, the data and control plane of the SDN architecture requires another dimension of decoupling between service functions and network infrastructure to manage them [1] dynamically.

The existing literature has attempted to address these challenges using network functions virtualisation (NFV), which refers to cloud networking concepts and applies this concept in the telecommunications space to deploy network functions as virtualized software instances instead of dedicated hardware appliances. This enables the creation of logically isolated network partitions over a shared physical network infrastructure, allowing the aggregation of multiple resources as single resources (Papneja, 2022). Researchers have combined SDN with NFV to deliver service-oriented network functionalities (Papneja, 2022). However, the majority of the work is at a theoretical stage. This leaves the objective of guaranteeing end-to-end QoS provisioning in SOA-based SDN an open question.

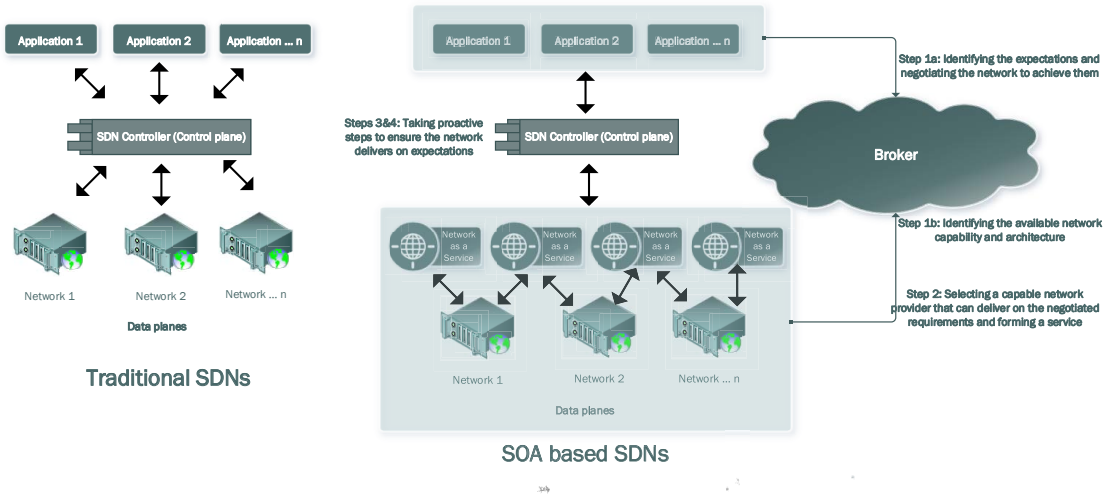


Figure 2.1: Difference between traditional and SOA-based SDNs (Khan et al., 2021)

In the networking context, QoS measures a network's capability in terms of different parameters while delivering it as a service. Specific to computer networks, QoS can be measured in other parameters such as inadequate bandwidth, end-to-end traffic delay, jitter, and packet loss (Tomovic et al., 2014). Guaranteeing end-to-end QoS provisioning means ensuring that the delivered network meets the negotiated expectations. The Cisco Internetwork Operating System (IOS) defines three types of architectures: best effort, integrated and differentiated, which provides a network with different level/s of guarantees. A best-effort model is where all the packets receive the same priority, there is no guaranteed delivery of packets, and the network delivers data based on its capability without any assurance of reliability, delay bounds, or throughput (Ford et al., 1997). A best-effort service is suitable for a wide range of applications where data delivery assurance is not mandatory, such as general file transfer or emails (Cisco Systems, 2014) (Cisco Systems, 2017). An integrated service architecture commits to meeting the defined QoS requirements of the application. This architecture requests a network of specific requirements before sending the data (Cisco Systems, 2014). These architectures are suitable for video or sound applications and must be delivered without interruption. A differentiated service architecture provides a scalable network with differing level/s of QoS as required by the application (Cisco Systems, 2014). Such architectures are mainly used for mission-critical applications. Depending on the necessary level of scalability, a differentiated network can further be classified as either *hard* or *soft QoS* types. Hard QoS is the "absolute reservation of network resources" (Fachrunnisa, 2011). Soft QoS reservation of network resources, on the other hand, is not firm(Fachrunnisa,

2011). In such a case, the network provided may not meet the defined requirements for a period. These architectures are available when autonomous applications use SOA-based SDN to request network capability for their requirements. However, according to the characteristics of these architectures, not all of them will suit the needs of an application. Thus, to guarantee the end-to-end QoS provisioning in SOA-based SDN, an essential requirement for the requesting applications is to follow a systematic process that will assist them in negotiating, selecting, monitoring, and managing the correct type of network with providers which are capable of delivering on their specific requirements. If this process is not followed, there is a very high chance that the provided network QoS will not meet the needs. As shown in Figure 2.1, incorporating this process to guarantee the QoS of SOA-based SDN requires a paradigm shift in how the Network as a Service (NaaS)is formed and managed compared to how it is done in a traditional SDN model.

The motivation of this review paper is to identify and discuss the existing QoS-guaranteed approaches in SDN and analyze if they meet the specific requirements needed for forming informed networks between autonomous services in SOA-based SDN. The structure of the review paper is organized as follows: Section 2 identifies the essential requirements to guarantee end-to-end QoS provisioning in SOA-based SDN architecture. Sections 3-7 discuss the QoS management in SDN using the controller design-based approach, the dynamic resource allocation approach, the queue scheduling-based approach, the optimal routing approach, and the SLA-based quality management approach, respectively, and their limitations in providing end-to-end QoS guarantees in SOA-based SDN. Section 8 summarizes the challenges and outlines the directions for future research.

2.3 Key Requirements to Guarantee End-to-end QoS Provisioning in SOA-based SDN Architecture

To identify the critical requirements in guaranteeing end-to-end QoS provisioning in SOA-based SDN, we take inspiration from cloud computing, one of the successful computing

architectures that have arisen from SOA. In its simplest form, cloud computing offers computing and storage resources on-demand as a service to its users. This enables the users to use these resources on a PAYG model rather than spending vast amounts upfront to acquire them physically. This paper focuses on how cloud services between users and providers are formed. In this process, researchers have identified many criteria that range from certifications to migration support, which must be considered when deciding on a service (Cloud Industry Forum 2019). One such measure, which is our focus in this paper, is forming SLAs with service providers who are reliable and who can deliver on the needed expectations.

SLAs are contracts that are agreed upon and signed between a service provider and another party, such as a service consumer, broker agent, or monitoring agent (Alhamad et al., 2010). The managing SLA aims to ensure that the defined services meet a certain level of criteria that has been restricted in the agreement. The SLA contains the terms of the services and includes the non-functional requirements of the services. These are known as the (QoS terms which include the obligations to be met, service pricing, and penalties in case of agreement violations (Emeakaroha et al., 2010). Failure to meet the conditions of the agreement may result in SLA degradation or SLA violation (Hammadi & Hussain, 2012). The occurrence of service degradation or service violation may have a significant impact or loss on the total business outcomes. Research on this criterion comes under the broad area of cloud service selection, which ranges from service discovery (Afify et al., 2014), service negotiation and selection (Z.-u. Rehman et al., 2015), proactive service management to preventing SLA violations (Hussain et al., 2018), service migration, if needed (Z. u. Rehman et al., 2015) and lastly ranking service providers according to their performance (Garg et al., 2013). The literature discusses that having such measures is beneficial to service users and service providers. From the service provider's perspective, it enables the creation of healthy competition (Garg et al., 2013). From the service user's perspective, it allows them to make informed and intelligent service-based decisions that will enable them to achieve their expectations in a decentralized environment (Hussain et al., 2017). In other words, the cloud service selection process does not follow a one-size-fits-all approach for all the service selection decisions. Still, it follows a customized strategy that first identifies the available services, negotiates between the service providers and the service users to form expectations, forms an SLA based on those expectations, and then proactively manages them to ensure that the QoS of the service delivered matches these expectations, before ranking

the service provider on a scale. Similarly, it is our view that if an SOA-based SDN needs to provide a network that satisfies the requirements of the different services, it must follow a similar approach that would first ensure the correct identification of services according to the needs of the application, forming SLAs with them, and then managing them to ensure that the expectations are achieved. To achieve such an approach to guarantee end-to-end QoS provisioning in SOA-based SDN architecture, the following requirements need to be met.

2.3.1 Ability to Personalise the QoS Required from a Service (R1)

Personalisation helps meet an application's (or consumer's) demands more effectively and efficiently, where the 'one-size-fits-all' approach is not applicable. In SOA-based services, personalisation is a feature that can be used during service selection and is beneficial to both the application users and network service providers. To the application users, it assists in guaranteeing that the network to be delivered will meet their requirements, and they will only pay for the network they use. To the network service providers, it assists in knowing the QoS to be delivered and deciding if they have the required capability to form an SLA with the specific application. Personalisation is also termed negotiation, after which the service requirements are demarcated in the SLA (Shuraia Khan, 2019). While the benefits of personalisation or negotiation have been applied in various domains, such as cloud computing, it has not received the same type of exposure to personalised network services in SDN. As a result, there are limited opportunities to receive personalised QoS delivery in SDN based on an application's needs, which is one of the requirements to deliver and use resources using an SOA-based network.

2.3.2 Reputation Value-based Network Service Selection (R2)

In the consumer market, the reputation value of a service or product is widely used to make a fair judgment on it. This is especially beneficial when there are many possible options available, and the consumer wants to select the one that best matches their needs. A higher reputation value of a provider represents a higher level of service satisfaction from the users,

which results in more trust among potential consumers. Offering reputation-based network service selection in SDNs will enable the requesting applications to make an informed decision as to which service provider to choose based on the QoS requirements. These need to be considered when the network is being delivered as a service, based on the reputation of the provider who is committing to deliver what it has promised.

2.3.3 Measuring the Satisfaction of a Network Service Provider Based on the QoS it Provides (R3)

In order to achieve requirement 2, one key aspect is to measure the satisfaction rating of the service providers. This should be determined by comparing the QoS values that the SDN service providers are committed to providing with their services defined in the SLAs (also termed QoS guarantees) against the actual QoS values they deliver (also termed *QoS actual delivery*). Furthermore, the satisfaction rating should not be subjective and ambiguous and should be determined on a globally agreed scale by a commonly agreed process (DR, 2013). This is important as SOA-based SDNs can be requested as a service by different geographically diverse users. Hence, if each application determines the satisfaction value using its own method, it does not accurately represent a service provider's capability to deliver on the promised QoS. The existing literature on SOA-based SDN does not have a mechanism to measure and quantitatively describe the satisfaction of the network service provider on a standardized scale, nor does it have a standardized process to do this. Without this, the service provider's satisfaction which represents an important criterion during the network service selection stage, cannot be considered.

2.3.4 Measuring the Network Provider's QoS Actual Delivery by Considering its Composite Nature (R4)

Given the flexibility of SDNs, providers offer various service guarantees such as network reliability, scalability, and performance under latency limitations, which come under application-based and network-based categories. Researchers such as (Akella & Xiong, 2014;

Karakus & Durresi, 2017) have studied various QoS guarantees. However, to facilitate requirements 2 and 3, the actual QoS delivery values of each service provider should be represented in each aspect of QoS measurement metrics and not just as a composite measure. This is important to differentiate the network providers in terms of their QoS. This concept has been utilized in cloud service selection; service users use multi-criteria decision-making (MCDM) methods to rank service providers according to their specific requirements and then choose the best one (Garg et al., 2013; Nawaz et al., 2018; Rehman et al., 2012; Youssef, 2020). To facilitate and ensure that SOA-based SDNs are used according to their potential, a composite representation of QoS provided by each network provider, along with the QoS measure provided in each aspect of QoS parameters, needs to be represented for each SDN. Furthermore, to ensure that SOA-based SDNs can be requested as a service by different geographically diverse users, QoS guarantee values should be maintained in a QoS repository that is globally accessible.

These four requirements (Req:1-4) form the basis of our investigation into the existing approaches utilized for guaranteeing QoS in SDNs. Our objective is to identify if the current QoS management approach proposes a solution to achieve these requirements that will ensure SDN is delivered as a service in an SOA-based model. A structure of the literature review is presented in figure 2.2 below.

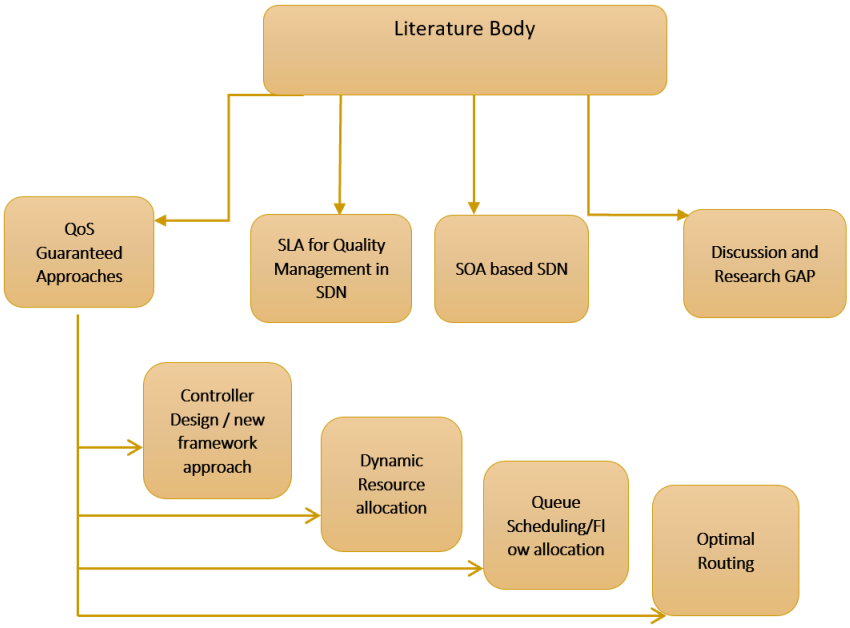


Figure 2.2*: Structure of the Literature review* by S. Khan, 2022.

We categorize the existing systems to achieve QoS in SDNs into five groups: *controller design-based QoS management*, *dynamic resource allocation approach for QoS guarantees*, *queue scheduling-based QoS management, optimal routing approach for QoS guarantees,* and *SLA-based quality management.* The following sections introduce these approaches for QoS management to determine if they address the requirements needed to guarantee QoS in SOA-based SDNs.

2.4 Controller Design-based QoS Management

Controller design-based QoS management enables network managers (controllers) to configure and manage the entire network from a central location. An SDN's control plane encompasses one or more software-regulated SDN controller(s) that enable it to control and manage network functionality by administering the traffic forwarding behaviour through a Controller-Data Plane Interface (C-DPI). Functional components and control logic are the two main elements of the SDN controller. The functional component manages the control behaviour, controls the logic component, and maps the network requirements for application demand and network

element resources accordingly (Karakus & Durresi, 2017). As the controllers can obtain the global view of the network, they use this to apply control policies to the network to manage the SLAs. Researchers have used the specifics of the architecture and modified it to propose new designs for guaranteeing QoS in SDN. The core concept of this approach is modifying the control logic component to implement policies based on SLA specifications or application or user demands. The functional component brings out the controller behaviour and network behaviour based on the response of control logic components.

[27] proposed a controller reference architecture named Q-Ctrl to achieve QoS in SDN-based cloud infrastructure. Q-Ctrl conjoins with another QoS lifecycle model to conduct effective QoS operations in SDN. It aims to achieve end-to-end QoS for enterprises, multimedia, and scientific applications and execute them in a virtual overlay network via Open vSwitch (OVS) to work in either direction, controller, or simulator mode. In this approach, QoS requests are scheduled to the QoS flow injector to inject the required QoS flows into the open flow-enabled networking devices. The QoS lifecycle proposed by researchers is classified into five major operation tasks: queue creation, QoS flow addition, QoS flow modification, QoS flow deletion, and queue deletion. Q-Ctrl has been tested on web and video streaming applications, and the results show that the allocation of bandwidth has been regulated effectively. The controller placement problem (CPP) is one of the significant challenges in achieving QoS in SDN (Jadhav et al., 2017). Based on the locations of switches, CPP is defined as choosing suitable locations for "K" controllers in a given network that could minimize the latency between controllers and switches and attain maximum performance metrics such as propagation delay, network reliability, load distribution, and failure resilience. (Cheng et al., 2015) studied a QoS-guaranteed CPP solution where the main focus was on placing the nominal number of controllers in the network that can minimize the response time of the controllers. Therefore, two heuristic algorithms (primal-dual-based algorithm and network partition-based algorithm) have been proposed and tested on the Internet Topology Zoo database (a database that includes publicly available network topologies) in a simulation environment. The results show that the Incremental Greedy Algorithm slightly outperforms the other two. However, the network partition algorithm incurs less response time.

Other research (Egilmez et al., 2012) proposed a novel open flow controller design-based framework named OpenQoS for end-to-end QoS services on multimedia delivery over SDN. The proposed approach involves grouping the incoming traffic as data flows and multimedia flows, followed by dynamically placing the multimedia flows in QoS-guaranteed routes and the data flow in the traditional shortest path route. The proposed framework develops a QoS architecture and then runs the OpenQoS framework. It also creates a dynamic QoS routing-based optimization framework to manage multimedia traffic. To measure the performance, the researchers conducted experiments over a real SDN test environment and compared the performance results with the current state-of-the-art, HTTP-based multibit rate-adaptive streaming. The experiment results show that the OpenQoS-based approach can guarantee seamless video delivery with little or no video artifacts experienced by the end users. SDN protocols such as OpenFlow enable traffic control on a per-flow level which is one of the prerequisites for an end-to-end performance guarantee in SDN. (Tomovic et al., 2014) Proposed an SDN-based control framework for QoS provisioning that presents the original design of the SDN/OpenFlow controller environment and provides bandwidth guarantee for priority flows in an automated manner. This control framework enables automated and flexible control on network devices by programming the controller to achieve the required QoS level for multimedia applications. In terms of QoS provisioning, the researchers introduce several service models and mechanisms, such as Integrated Services (IntServ) / Resource Reservation Protocol (RSVP), Differentiated Services (Diffserv), and Multi-Protocol Label Switching (MPLS). They also execute centralized control monitoring and determine the overall state of the network resources to accomplish innovative traffic management based on the current network state. In addition, they also minimize the degradation of best-effort traffic by providing priority flows. To compare their proposed solution with traditional techniques, they utilize the best-effort service and Int-Service and demonstrate that their model outperforms the best-effort shortest path routing and IntServ. (Adami et al., 2015) proposed an approach where the network control application is built on top of the existing floodlight tool for end-to-end QoS provisioning in SDN. The researchers modified the Dijkstra algorithm to link cost change according to the traffic load. By considering the network status as QoS metrics and ensuring no excessive devices or links resulting in slow transmission and packet loss are attached, they developed a system that guarantees QoS for specific traffic groups. The experiment on the emulation environment demonstrates the system behaving as expected and managing network resources more efficiently, thereby providing guarantees on traffic handling.

To summarise, the controller design approach is one of the popular approaches for SDN management, which has been utilized in different applications that range from providing a QoS-based solution to the entire SDN network to providing application-based solutions such as multimedia applications. However, as shown in Table 2.1, the controller design approach has drawbacks that do not commit to the requirements of SOA-based SDN to provide an end-to-end QoS guarantee.

Study	Algorithms/ Architecture	Contributions	Area of Applications	Implementation	R1	R2	R3	R4
(Govindarajan et al., 2014)	Q-Ctrl model with QoS lifecycle (Bandwidth Allocation)	Effective	Web and Video Streaming	Simulation & Real Network	×	×	×	×
(Cheng et al., 2015)	Incremental greedy, Primal-dual based, Network partition-based	-Incremental Greedy Outperforms, -Network partition less response time	Global	Simulation	×	×	×	×
(Egilmez et al., 2012)	OpenQoS framework, Dynamic QoS routing	Guarantee seamless video delivery	Multimedia	Real Network/ Test Bed	×	×	×	×
(Tomovic et al., 2014)	Per-flow control, automatic priority flows	-Automated Bandwidth guarantee for priority flows	Multimedia, Other applications include monitoring, control of resources	Real Network/ Test Bed	×	×	×	×
(Adami et al., 2015)	Modified Dijkstra Algorithm	-More Efficient Resource Management, -Guaranteed traffic handling	Global	Simulation	×	×	×	×

Table 2.1: Controller design or controller placement approach for the QoS guarantee of software-defined networking

2.5 Dynamic Resource Allocation Approach for QoS Guarantees

Resource allocation in the computer networking context is a process of assigning and managing the available network assets or resources in an optimized manner to support the network's demand. The resource allocation approach is also known as dynamic resource allocation for priority users or the resource reservation approach. QoS guarantee resource allocation and provides the corresponding network services for some applications to satisfy users' demands, such as performance metrics, latency, and bandwidth. Researchers have used resource allocation approaches in several contexts to achieve QoS in SDN networks, especially in multimedia applications. SDN provides the opportunity to improve the ability to guarantee the QoS of the Internet by arranging apposite resource allocation and scheduling mechanisms (Hu et al., 2015). Using such functionality, researchers have developed an innovative framework, named SoIP, for dynamic network management and allocating resources. The proposed model enhances the capability of the QoS guarantee on the internet by building a software-defined overlay network over the IP network. This technique brings the advantage of per-flow management characteristics in SDN to satisfy the resource demand of the aligned applications. This framework also represents the resource scheduling mechanisms for QoS guarantees in SDN-based overlay networks that are able to achieve the seamless combination of SDN and IP networks. The proposed model was tested in an emulation environment. The results demonstrate that SoIP is able to ensure the QoS requirements of end-users and applications on performance metrics.

Failure recovery is one of the critical challenges service providers face while achieving QoS in a network. Sharma, Figueiredo, et al. (2014) proposed a QoS framework using a vendor-agnostic interface of SDN technologies such as OpenFlow, OF-Config (OpenFlow configuration and management protocol), and OVSDB (Open vSwitch Database Management Protocol). The focus of these approaches is that high-priority traffic should always be given higher precedence over best-effort traffic. The framework was tested on a single autonomous system (AS) (emulated pan-European topology) and multiple AS scenarios (designed in the City Flow project on the OFELIA testbed facility in iMinds) evaluated in three failure recovery scenarios. This approach reserves resources for the high-priority flow of entrance switches, whereas Open QoS does not use any resource reservation scheme. The mechanism of this framework consists of several steps that start with configuring three default queues on each port using the OVSDB protocol. The controller then runs standard routing protocols (OSPF,

BGP) on each router using Route Flow, followed by establishing flow entries in the router. The aim is to configure and ensure the availability of network resources for high-priority traffic with a rate-limiter queue. The experiment results show that high-priority traffic can gain precedence over best-effort traffic even in failure conditions. The results show that the proposed framework is suitable for the future internet using SDN accommodating operations.

OpenFlow networks/SDN provide bandwidth guarantees using well-known FIFO scheduling. As a result, OpenFlow switches might not meet the QoS requirements for some applications, such as multimedia streaming. This limitation creates a packet scheduling issue. (Ishimori et al., 2013) Proposed a QoS development strategy (QoSFlow) for OpenFlow to enhance the control of multiple packet schedulers of the Linux kernel. The framework comprises a QoS module added to the standard OpenFlow data path. The QoS module consists of three components, namely traffic shaping, packet schedulers, and enqueuing. OpenFlow 1.0 with a FIFO scheduler was used to achieve different treatments for the packet. The strategy was tested in a test environment where an Open WrtBackFire was installed according to Pantou projects. The experiment was evaluated on several parameters, such as response time, switch capacity, number of queue impacts, bandwidth isolation, and QoE evaluation. The evaluation results show that the proposed approach has a low response time, and the use of Stochastic Fairness Queuing (SFQ) brings improvements (an increase of more than 48% on the PSNR value) on QoE. (Akella & Xiong, 2014) Proposed an approach to network resources using cloud carriers upon users' requests and subject to QoS requirements. Jeong et al. (2012) proposed a QoS-aware network operating system for SDN service provisioning using generalized open flows. Kassler et al. (Kassler et al., 2012)(2012) proposed a system to enable negotiation among service network communications between end-users and allocated network paths for sending multimedia flows according to the service configuration as promised. This system has the capability of quality of experience (QoE)-driven path allocation and optimization for multimedia services in SDN. A multi-tenancy management-based framework was proposed by(Porxas et al., 2015) to ensure QoS in SDN through tenant isolation, prioritisation, and flow allocation. This management framework offers a virtualization technique (QoS-aware virtualization-enabled routing (QVR)) which can be implemented in the SDN architecture and combined with the proposed QoS-aware framework to allow flow allocation concerning different tenant applications. Karaman et al. (2015) analyzed a high-quality, uninterrupted

Voice over Internet Protocol (VoIP) service accompanied by a video of users using the SDN. This research experiments on allocating QoS for regular users trying to communicate with one another in a congested network with background traffic. Argela SDN controller is employed to manage QoS for different flows. The controller communicates with the Argela SIP server to receive the SLA parameters for each VoIP subscriber and makes QoS decisions based on these specified parameters. The experiments in this study are conducted on an output of the conjoint works of the OFERTIE and SIGMONA projects. The results showed that the flexibility enabled by SDN improved the QoS of the network with a reduction in loss to less than 5% and a more than 50% reduction in latency and jitter.

A framework for applying the service-oriented principle, NaaS, in SDN was proposed by(Duan, 2014) to address the challenging issue of end-to-end QoS provisioning. The critical function of end-to-end QoS provisioning is allocating sufficient bandwidth in network services. A NaaS in the SDN framework sits on top of an SDN controller or a set of SDN controllers. A NaaS abstraction interface is between the SDN controller domain and the orchestration module to abstract the network functionalities of each part and is named the network service. The orchestration module takes the service requests from the upper-layer application module, determines the required amount of bandwidth for each active network service, and then acquires the bandwidth to achieve given end-to-end QoS requirements. The arrival curve concept in network calculus is employed in this framework to develop a general abstract load profile. Analytical and numerical results show that this framework is able to support end-to-end QoS with flexible service management and higher bandwidth utilization.

To summarise, the dynamic allocation of network resources is one of the practical approaches to achieving QoS delivery in SDN networks. A comparative analysis of the techniques proposed to achieve this aim is shown in Table 2.2. This comparative analysis clearly identifies that such approaches are preferable for multimedia applications where the availability of network resources is crucial. However, these approaches do not guarantee the provision of QoS delivery in a global network. Furthermore, these approaches cannot provide personalised service delivery for the end user. Most of the techniques have been evaluated in a simulation

environment; as a result, there are fewer opportunities to understand how the methods will behave in a real network and the SOA-based SDN environment.

Study	Algorithms/ Architecture	Contributions	Area of Applications	Implementation	R1	R2	R3	R4
(Hu et al., 2015)	SoIP framework (resource allocation)	Potential	QoS Guaranteed Internet	Simulation & Real Network	×	×	×	×
(Sharma, Staessens, Colle, Palma, Goncalves, Figueiredo, et al., 2014)	OpenFlow, OF-Config, OVSDB	-Suitable (Future Internet) -Priority traffic gets precedence	Global	Simulation	×	×	×	×
(Ishimori et al., 2013)	OpenQoS framework with FIFO schedulers	-Low response time -48% on QoE improvement	Multimedia	Test Bed	×	×	×	√
(Jeong et al., 2012)	Per-flow control, automatic priority flows	-Automated Bandwidth guarantee for priority flows	Multimedia & Other	Simulation	×	×	×	×
(Porxas et al., 2015)	Virtualization-enabled Routing (QVR)	-More Efficient Resource allocation	Global	Simulation	×	×	×	×
(Karaman et al., 2015)	Resource allocation	Loss reduced	VoIP	Test Bed	×	×	×	√

Table 2.2: Comparative analysis of dynamic resource allocation-based approaches for QoS guaranteed service delivery in SDN

2.6 Queue Scheduling-Based QoS Management

The packets are processed in a queue that follows the first-in-first-out (FIFO) queue processing rules. However, some packets in the queues require a higher priority for processing than other packets ahead of them. This queuing technique has a significant impact on QoS service delivery along with traffic shaping (Karakus & Durresi, 2017). In addition, multiple packet schedulers require a more optimal queuing approach to maintain the required network performance. The queue scheduling technique maps a packet to an internal forwarding queue based on QoS requirement information and drives the queues according to a predefined queuing policy.

Li et al. (2017) proposed queue scheduling-based QoS techniques for cloud applications in SDN. Their approach identifies applications and determines the required QoS levels for each application type. It then implements a queue scheduling technique to permit delay-sensitive data to be de-queued and sent first. Additionally, multi-queue techniques have been set up in each output interface of the switch, such as the expedited forwarding (EF) queue with the highest priority, the assured forwarding (AF) queue with medium priority, and the best effort (BE) queue with the lowest priority. There are three main modules to control and manage messages in the system design stage. The control message module forwards the notices according to the flow table rules, whereas the queue management module configures queues based on configuration information. The queue scheduling module schedules the packet out of the lines with different priorities. An evaluation of this approach was conducted using both experimental and theoretical analysis. The theoretical analysis shows that this method can provide differentiated services for application flows and is able to map to different QoS levels. The experiment results show that delay can be reduced by 28% on average when the output interface has sufficient available bandwidth. In addition, in terms of the highest priority for application flow, this method can reduce the delay by 99.99% and increase throughput by 90.17% on average when the output interface utilization reaches the maximum bandwidth limitation.

(Caba & Soler, 2015) proposed an application programming interface (API) named QoS Config API for configuring QoS resources dynamically by allowing applications to configure priority queues to the ports of network devices. The OVSDB protocol was used at the Data-Controller Plane Interface (which enables the entire network to be programmed) of an existing SDN controller (SDNC) to implement the proposed API. The implementation was conducted on a distributed testbed. The first test demonstrates that the implemented QoS Config is able to offer control dynamically and allocate bandwidth to different traffic flows in the data plane using a rate-limiting queue. Moreover, there is a possibility of creating high-level abstractions from APIs exposed by the SDNC. The second test demonstrates that QoS Config API's current performance is sufficient to enable the dynamic configuration of QoS. The SDNC reactively installs flow entries. As a result, it is possible to perform QoS configurations with little impact on the flow setup delay. Sharma, Pickavet, et al. (Sharma, Staessens, Colle, Palma, Goncalves, Pickavet, et al., 2014) developed a QoS framework for OpenFlow by configuring the business

consumer traffic to a high-priority queue in all conditions (such as the network failure condition). This approach divides the traffic into business and best-effort traffic, configuring priority queues and redirecting different flows to suitable priority queues. The OpenFlow protocol with the OVSDB configuration protocol provided high QoS. The experiments were performed in an emulated OpenFlow environment named the OFELIA testbed facility provided by iMinds. The protocol was tested on stream video clips in an emulated OpenFlow pan-European topology. The results show that business consumers achieve a higher quality of service than best-effort consumers. Upon experiencing failure, this framework can re-establish flow entries on the affected paths, and the edge router is able to reconfigure its rate limiter queues appropriately for network traffic when applying dynamic QoS mechanisms in OpenFlow-enabled SDN switches. The measurement study is accompanied by two fundamentally different QoS techniques named priority queue and bandwidth guaranteeing queue. The results reveal a noticeable variation for separate OpenFlow switches in terms of performance.

Other queue implementations, such as the FIFO or SFQ queues, significantly impact network performance. In terms of bandwidth guarantees, one hardware switch violates the configured bandwidth guarantees. Other than that, for all setups, deploying dynamic QoS mechanisms leads to duplicated TCP packets, wasting network resources. Table 2.3 shows a comparative analysis of queue scheduling approaches concerning the requirements for guaranteeing end-to-end QoS provisioning in SOA-based SDN architecture.

Study	Algorithms/ Architecture	Contributions	Area of Applications	Implementation	R1	R2	R3	R4
(Li et al., 2017)	Application Identification, Queue Scheduling	Differentiated services.	Cloud Applications	Real Network	×	×	×	√
(Caba & Soler, 2015)	Rate limiting Queue	Dynamic QoS Configuration, Fine Granular service	Global	Distributed Test Bed	×	×	×	×
(Sharma, Staessens, Colle, Palma, Goncalves, Figueiredo, et al., 2014)	OpenFlow and OVSDB	-Restabilised the OpenFlow after failure	Global	Test Bed	×	×	×	×
(Durner et al., 2015)	Dynamic QoS Mechanism	Different Queue significantly impacts network performance.	Global	Real Network/ Test Bed	×	×	×	×

Table 2.3: Comparative analysis of existing queue scheduling approaches for QoS-guaranteed software-defined networking

2.7 Optimal Routing Approach for QoS Guarantees

Finding the best route that provides guaranteed QoS for flows is the ultimate challenge in a network. Calculating the most efficient route without delay is not easy since network resources can dynamically change at any time. In addition, routing in today's traditional networking is a complicated issue due to many unsolved challenges, such as the network's global view limitation, per-hop decision-making, and applying flow-based QoS (Karakus & Durresi, 2017). Moreover, the propagation of high resource-demanding applications, such as video conferencing, VoIP, etc., on the internet requires a more sophisticated, dynamic, and efficient routing mechanism that is able to meet the QoS demand. QoS-driven routing is able to provide routing strategies that are capable of identifying the best paths to satisfy the maximum possible number of flows with QoS requirements. As a result, QoS-driven routing is needed to keep flows under QoS-guaranteed routes and provide optimized QoS.

An SDN-based QoS control model for fast routing in an industrial QoS network was proposed by (Guck et al., 2015). This model avoids control loops over forwarding and control planes. However, it can route flows through multiple queue links with different QoS levels. Thereby maintaining QoS through delay constrained least-cost (DCLC) routing is developed as an online algorithm for admission control based on the existing DCLC routing algorithm. The proposed control model has been compared with a mixed integer programming (MIP)-based approach in a simulation environment. The results show that this model-based approach can make accurate routing and admission control decisions within a few microseconds. This achievement utilizes the link average up to approximately 93% in an industrial communication scenario compared to close to 100% utilization by the MIP approach. (Chen et al., 2017) proposed a QoS-guaranteed systemic approach for dynamic service chaining in SDN. Service

chaining is the deployment procedure of building a sequence of individual service functions to accomplish a complex task. The authors developed an analytical method using network calculus (NC) and queuing theory to determine the delay performance and characterize them. They also developed an optimal path determination algorithm for service chaining with a minimal workload. The experiments were conducted on real and virtual networks using a Mininet simulation environment, demonstrating that NC delay analysis can provide deterministic QoS-guaranteed service chaining for any satisfied delay requirements. This approach is also able to design a network for the future delay-sensitive internet where deterministic latency must be guaranteed. (Alharbi & Fei, 2016) Proposed a QoS-based framework for the smart grid using SDN that dynamically allocates critical flows. The authors divide the data traffic into best-effort traffic and critical traffic. Best-effort traffic does not have QoS requirements, and critical traffic has one or more QoS requirements. The control program monitors the status of the network and redirects the critical flows over a better path by installing OpenFlow rules in the switch. The researchers developed a path-searching algorithm for the QoS path and implemented it as a module in the floodlight controller. A Mininet emulation environment was used to test the QoS routing algorithm. The results demonstrate that, compared with the shortest-path routing algorithm, this approach improves the overall performance obtained by critical flows, with the network achieving a higher utilization level. (Hongyu et al., 2015) proposed a QoS-guaranteed SDN-based routing strategy aimed at saving energy in the backbone network. The proposed technique integrates the backbone networks' energy-saving strategy (BNESS) with the Open Shortest Path First (OSPF) protocol with the maximum clique problem (MCP) to search idle links to save energy. The approach was tested using a Mininet simulation environment, and the results show that the BNESS algorithm is simple and can save energy effectively.

Akella and Xiong (Akella & Xiong, 2014) proposed an approach for all priority cloud users in SDN by allocating bandwidth to meet the QoS requirements. This approach presents an efficient QoS routing algorithm by considering congestion, available bandwidth, and hops count. A mathematical expression was developed for path selection to meet the QoS requirements in the cloud to attend to multiple cloud users. This proposed QoS guaranteed approach consists of (a) a new metric based on available bandwidth, path length or hop count, and Round Trip Time(RTT) and (b) the queuing technique or policies for multiple cloud users.

A greedy algorithm was used for path selection. This proposed algorithm can automatically switch available paths for higher-priority clients to ensure the guaranteed QoS. The Utah Emulab testbed was used to evaluate the performance of the proposed QoS guaranteed routing approach, and the results show the approach's effectiveness. (Seliuchenko et al., 2016) proposed a multi-commodity flow allocation-based model to provide optimal routing based on QoS criteria. The proposed routing model balances the load based on the minimum-maximum load of network channels and the service quality of each stream. A flow identification technique was also proposed to find the optimal set of routes through the network for all flows with minimal total cost. The optimization finds the shortest path for high-priority flows and distributes the low-priority flows to align the network load. The experiments on this model were configured in the Mininet emulator, and the results show that the delay for the first-class flows decreased by 20%, and the wait for non-real-time redirected flows increased by 10%. However, in both cases, the average delay was below the critical threshold. Table 2.4 shows a comparative analysis of QoS-guaranteeing approaches regarding the requirements to guarantee end-to-end QoS provisioning in SOA-based SDN architecture.

Study	Algorithms/ Architecture	Contributions	Area of Applications	Implementation	R1	R2	R3	R4
(Guck et al., 2015)	-Online Algorithm on DCLC routing	-Accurate Routing -Admission Control Decision in Micro Second	Industrial	Simulation	×	×	√	√
(Chen et al., 2017)	-Network Calculus and Queuing Theory -Path Determination Algorithm	-Deterministic QoS Guarantee	Global	Real Network and Emulated Environment	×	×	×	×
(Alharbi & Fei, 2016)	Path Searching Algorithm for QoS Path	-Performance Improved, -Higher-level Unitization	Smart Grid	Emulation	×	×	×	×
(Hongyu et al., 2015)	-OSPF with Maximum Clique Problem	It-Simple and can save energy	Backbone Network	Simulation	×	×	×	×
(Akella & Xiong, 2014)	-Matric-based path selection -Greedy Algorithm -Queuing Technique	Effective	Priority Cloud Users	Emulab Test Bed	×	×	×	×
(Seliuchenko et al., 2016)	-Flow Identification technique	Delay is below the Threshold	Global	Emulation	×	×	×	√

Table 2.4: Comparative analysis of QoS-driven routing approaches for the QoS-guaranteed software-defined network.

2.8 SLA-based Quality Management

A policy-based management (PBM) approach was proposed for managing the QoS in an SDN (Machado et al., 2014). This approach introduces an automatic policy refinement technique that can translate the SLA terms and conditions into a set of corresponding low-level rules. The high-level policies (SLA) are translated manually into technical requirements or service-level specifications (SLS) and then translated into possible service-level objectives (SLO). Each value of these objectives is obtained by querying a specific QoS class in an LDAP repository. The administrator previously configures the repository with a set of QoS classes (platinum, gold, silver), class Requirements (delay, jitter, and bandwidth), and the requirement value (200ms, 20ms, and 128 kbps). LDAP also contains a protocol list. The administrator sets the protocol H.323 in the platinum class registered in the LDAP repository. The controllers receive the rules from the repository that is filtered by categories. Then, the controller analyses the network flow and finds the best path to enforce the rules. This approach is not tied to any specific controller. The authors conduct the experiments in the Mininet emulation environment. The results show that the proposed approach can quickly reconfigure particular rules as the researchers have already set up a populated list of the best links between switches. (Bhattacharya & Das, 2013) proposed a dynamic SLA that is capable of providing QoS-enabled network architecture to address QoS issues across the internet. This proposed architecture facilitates interaction among Internet service providers, which helps to build dynamic relations to ensure flexibility and adaptability. A single network consists of four network providers (NPs) and one service provider (SP) domain where a source (SRC) and a destination (DST) relate to the SP. A periodical database is stored by the SP. When SRC wants to have a session with DST, SRC sends a request to SP. The SP of the source asks the SP of the destination to find the current destination network. By querying the database, SP knows all the NPs involved in reaching the destination. NP also offers the best available path with QoS requirements. The evaluation using a simple scenario shows that the architecture can find the best way before the link breaks and the best alternative route after the link breaks.

A real-life implementation-based technique in an SDN is proposed by(Körner et al., 2014), which enables the application of QoS requirements by combining the WS-Agreement standard and the OpenFlow standard. This approach enables the following capabilities in the network: defining QoS requirements in virtual machines in the network, negotiating the SLO according to current network utilization, and creating an SLA for a cloud-based network, followed by establishing a QoS overlay in the OpenFlow network based on the SLA. These capabilities are achieved by implementing the following four subtasks: a) creating service description terms (SDT) which is an agreement that describes a part of or the complete service. Therefore, the involved parties understand the content of the SDT; b) developing an SLA framework named WSAG4J that implements the WS-agreement and the WS-agreement negotiation as the protocol; c) implement the OpenFlow protocol to define the flows in the network, make the forwarding decision and manages the queues and d) Implementing the Floodlight controller which is an open-source controller that helps the module discover the topology or forwarding path calculation. In addition, a cloud-based middleware was implemented to apply negotiation using the OpenFlow agreed quality constraints to the underlying cloud SDN substrate. The SLA4SDN prototype and the dynamically deployed QoS constraints are evaluated on an Open vSwitch-based emulation environment (Mininet). The experiment results show that this framework is able to manage the network in a much easier way as well as calculate the overall network utilization. Based on the calculation, the framework is able to predict the available capacity in relation to all end-to-end overlays.

Based on the aforementioned SLA-based QoS studies (Bhattacharya & Das, 2013; Machado et al., 2014) used the optimal routing technique to find the best path for network flow after receiving the service requirements through the SLA. However, neither of these studies provides an opportunity to select personalised services, nor do they have measures to receive a guaranteed level of QoS. In the approach discussed by (Körner et al., 2014), there is an opportunity to choose services and an ability to specify the QoS requirements and QoS level in the service negotiation stage. However, there are no measures defined for understanding QoS guarantees. (Körner et al., 2014) Identify the broad range of infrastructure as a service (IaaS) offers to outsource the IT infrastructure into the cloud. In contrast, the properties of the underlying SDN network and its connected servers are treated as an infrastructure resource.

There is no architecture detail or implementation detail available to achieve this service-oriented structure in SDN. Table 2.5 shows a comparative analysis of SLA-based quality management with respect to the requirements for guaranteeing end-to-end QoS provisioning in SOA-based SDN architecture.

Study	Algorithms/ Architecture	Contributions	Area of Applications	Implementation	R1	R2	R3	R4
(Machado et al., 2014)	An automatic policy Refinement Technique.	Translate SLA terms into a set of low-level rules. Finding the best path	Global, mostly internet	Simulation	×	×	×	×
(Bhattacharya & Das, 2013)	QoS enabled network architecture.	Dynamic relation to bringing flexibility and adaptability and path optimization.	Global	Simulation	×	×	×	×
(Körner et al., 2014)	Conjoin WS-agreement standard and OpenFlow standard approach	WSAG4J framework	Global	Real Network implementation	√	×	×	×

Table 2.5: Comparative analysis of QoS-driven service level agreement management-based approach for the QoS guaranteed software-defined network.

2.9 Discussion and Open Gaps from the Perspective of Guaranteeing End-to-end QoS Provisioning in SOA-based SDN Architecture

Based on the above discussion, QoS delivery in SDN is a prime research challenge for SDN researchers and developers to ensure users' satisfaction. While existing approaches have attempted to address this issue, the SLA-driven approach and most queue scheduling approaches have provided QoS solutions for SDN networks rather than application or area-specific solutions. Moreover, very little research has been done to affirm the successful response rate after implementing their approaches. Most of the research has been tested in a simulation, emulation environment, or test bed environment and not in the real SDN network.

Furthermore, based on the comparative analysis, there are no approaches that can provide reliable personalised service delivery for service selection decision-making. This, therefore, causes issues in delivering reliable personalised services to the SDN consumer. Moreover, there is no unique measurement technique available in SDN that can authenticate the satisfaction level of the provided services. As a result, there is always a gap between the service provider and consumer in relation to service satisfaction information that can be considered for service provider selection decision–making which generates a lack of transparency among service providers and consumers.

Furthermore, as shown in Table 2.6, none of the previous research proposes SOA-based QoS delivery in SDN except (Bueno et al., 2013; Duan, 2014). As a result, the existing literature does not consider t personalised service selection and guaranteed QoS delivery of SDN in SOA-based architecture. Therefore, the vital research area of achieving QoS in SOA-based SDN has not been investigated.

QoS management approaches	SDN Global Solution	SOA based SDN	R1	R2	R3	R4
Controller Design or Controller Placement Approach	× (except (Adami et al., 2015; Cheng et al., 2015))	× (except (Duan, 2014))	×	×	×	×
Resource Allocation or Dynamic Resource Allocation (Priority Users) approach	× (except (Sharma, Staessens, Colle, Palma, Goncalves, Pickavet, et al., 2014)	×	×	×	×	× (except (Karaman et al., 2015))
Queue Scheduling and Management approach	Primarily Global (except (Li et al., 2017))	×	×	×	× (except (Li et al., 2017))	× (except (Li et al., 2017))
QoS-Driven Routing approach	× (except (Chen et al., 2017; Seliuchenko et al., 2016))	×	×	×	× (except (Guck et al., 2015)	× (except (Guck et al., 2015; Seliuchenko et al., 2016)
SLA-Based approach for delivering QoS	√	×	×	×	×	×

Table 2.6: A comparative analysis of all the existing QoS guaranteed approaches in SDN from the perspective of applying it in SOA-based SDN

The comparative analyses shown in Tables 2.1 – 2.5 also demonstrate that there is no unique QoS measurement standard available in SDN that can be adapted to SOA-based SDN

architecture. While some researchers have focussed on this issue, the approaches that have been developed need further work for SOA-based SDN to be applied in a production environment. SOA-based SDN as a technology aims to provide cloud networking features such as an abstracted pool/grid of resources and elasticity in the provision of SDN in a service-orientation method (Papneja, 2022). After reviewing and discussing the relevant existing research, the following noticeable shortcomings are identified that need to be addressed.

1. There is currently no reputation-data-driven SLA-based approach that ensures QoS in SDNs; however, such a reputation-data-driven SLA-based approach could be used to select a suitable SDN service provider based on consumer requirements. This shortcoming needs to be addressed by developing an SLA-based intelligent framework where decision-making intelligence drives through the reputation rating score of an organization in delivering the SLAs. SDN service requestors can use reputation values for selecting SDN service providers

2. There is currently no existing method that enables a service requester to personalize service delivery and receive QoS as promised in SDN services. To ensure personalised service delivery, each service requester as an interacting party needs to determine their service requirements at a granular level based on which the SLAs will be formed. The provision of a service provider who delivers service with personalised requirements gives the service consumer the impression that the service provider considers them a priority consumer and values their business (Khan & Hussain, 2019). This notion that trust and personalised service delivery can be achieved and maintained only by consistently delivering services according to the requirements that have been agreed upon has been considered in the literature. This can only be achieved if the service requirements can be determined by both parties in a granular manner prior to the services being delivered.

3. No service satisfaction measurement technique can verify the guaranteed level committed to each SLO defined in the SLA. To a service provider and service consumer, the term *reliable* explicitly illustrates that there is no chance or a minimal chance of violating the defined SLA. As previously explained, the violation of an SLA may involve significant financial penalties for both parties. To avoid these scenarios, both service providers and consumer companies demand guaranteed reliable service

delivery. The existing studies discuss QoS guarantees in subjective-based terms. This means that the QoS guarantee level has not been benchmarked, so it cannot be used as a standardized measurement technique. Therefore, there is a need to develop a unified approach that can explicitly measure the guarantee of the services quantitatively (objective-based). For this, we need to develop an approach that can compare the received service quality level against the service quality level that was defined in SLA. To address this shortcoming, an intelligent approach is needed that evaluates the runtime services and therefore benchmarks the currently provided services with the committed services.

4. Very little research has been done that provides an end-to-end QoS guarantee from the perspective of an application being delivered in the SDN network. Existing research demonstrates that most SLA-based approaches are targeted to deliver the services or improve the solutions for the entire SDN network rather than specific application-based development. However, when SDN architecture is provided as a service, the focus should be on application-based QoS. To achieve this, further research is required to overcome the drawbacks and other shortcomings that are illustrated here to make the best of it. The successful integration of SDN and NFV would partially address the above research shortcomings. The possible solutions may include implementing the combination prototype of SDN and NFV architecture and testing, followed by recording the newly implemented combination network performance and benchmarking with further research for improvement.

To enable service provisioning according to the current business needs, the identified requirements R1-R4 need to be addressed. Without this, the realization of SOA-based SDN will remain in its infancy. In our future work, we aim to address these issues that will, in turn, lead to providing QoS-guaranteed SDN service delivery.

2.10 Conclusion

This chapter reported on an extensive survey of the existing literature. In analyzing the existing literature, we identified four primary requirements to guarantee end-to-end QoS provisioning in SOA-based SDN. Then, we grouped the existing literature on QoS management under several categories based on the methods developed to achieve the objective. Following this review of various perspectives on QoS management, we then evaluated the approaches based on our predefined requirements. We performed a comparative analysis of every category of the approaches to determine how the approaches align with the end-to-end QoS provisioning in SOA-based SDN based on our identified requirements. Finally, we evaluated all the existing literature approaches critically and found four shortcomings. We then summarized the shortcomings of the reviewed works and the extant research challenges with associated future work directions. In the next chapter, we define the problem we intend to address in this thesis.

2.11 References

Adami, D., Donatini, L., Giordano, S., & Pagano, M. (2015). A network control application enabling software-defined quality of service. Communications (ICC), 2015 IEEE International Conference on.

Afify, Y. M., Moawad, I. F., Badr, N. L., & Tolba, M. F. (2014). Cloud Services Discovery and Selection: Survey and New Semantic-Based System. In A. E. Hassanien, T.-H. Kim, J. Kacprzyk, & A. I. Awad (Eds.), *Bio-inspiring Cyber Security and Cloud Services: Trends and Innovations* (pp. 449-477). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-43616-5_17

Akella, A. V., & Xiong, K. (2014). Quality of service (QoS)-guaranteed network resource allocation via software defined networking (SDN). Dependable, Autonomic and Secure Computing (DASC), 2014 IEEE 12th International Conference on.

Alhamad, M., Dillon, T., & Chang, E. (2010). Conceptual SLA framework for cloud computing. Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on.

Alharbi, F., & Fei, Z. (2016). Improving the quality of service for critical flows in smart grid using software-defined networking. Smart Grid Communications (SmartGridComm), 2016 IEEE International Conference on.

Bhattacharya, B., & Das, D. (2013). SDN based architecture for QoS enabled services across networks with dynamic service level agreement. Advanced Networks and Telecommuncations Systems (ANTS), 2013 IEEE International Conference on.

Bueno, I., Aznar, J. I., Escalona, E., Ferrer, J., & Garcia-Espin, J. A. (2013). An opennaas based sdn framework for dynamic qos control. 2013 IEEE SDN for Future Networks and Services (SDN4FNS).

Caba, C., & Soler, J. (2015). Apis for qos configuration in software defined networks. Network Softwarization (NetSoft), 2015 1st IEEE Conference on.

Chen, Y.-J., Wang, L.-C., Lin, F.-Y., & Lin, B.-S. P. (2017). Deterministic quality of service guarantee for dynamic service chaining in software defined networking. *IEEE Transactions on Network and Service Management, 14*(4), 991-1002.

Cheng, T. Y., Wang, M., & Jia, X. (2015). QoS-guaranteed controller placement in SDN. Global Communications Conference (GLOBECOM), 2015 IEEE.

Cisco Systems, I. (2014). *Cisco IOS Quality of Service Solutions Configuration Guide* (Vol. 18). American Headquarters, Cisco Systems, Inc.

Cisco Systems, I. (2017). *Cisco EasyQoS Solution Design Guide* (1.6 ed.). Cisco.

Cloud Industry Forum (2019). *8 criteria to ensure you select the right cloud service provider*. Retrieved 30/03/2019, from https://www.cloudindustryforum.org/content/8-criteria-ensure-you-select-right-cloud-service-provider

DR, B. B. (2013, 11th May). Satisfaction has a legal meaning. *The Customer Institute*.

Duan, Q. (2014). Network-as-a-service in software-defined networks for end-to-end QoS provisioning. 2014 23rd Wireless and Optical Communication Conference (WOCC).

Duan, Q., Ansari, N., & Toy, M. (2016). Software-defined network virtualization: an architectural framework for integrating SDN and NFV for service provisioning in future networks. *IEEE Network, 30*(5), 10-16.

Durner, R., Blenk, A., & Kellerer, W. (2015). Performance study of dynamic QoS management for OpenFlow-enabled SDN switches. Quality of Service (IWQoS), 2015 IEEE 23rd International Symposium on.

Egilmez, H. E., Dane, S. T., Bagci, K. T., & Tekalp, A. M. (2012). OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks. Signal & Information processing association annual summit and conference (APSIPA ASC), 2012 Asia-Pacific.

Emeakaroha, V. C., Brandic, I., Maurer, M., & Dustdar, S. (2010). Low level metrics to high level SLAs-LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments. High Performance Computing and Simulation (HPCS), 2010 International Conference on.

Fachrunnisa, O. (2011). *A methodology for maintaining trust in virtual environments*, Curtin University].

Ford, M., Stevenson, T., Lew, H. K., & Spanier, S. (1997). *Internetworking technologies handbook*. Macmillan Publishing Co., Inc.

Garg, S. K., Versteeg, S., & Buyya, R. (2013). A framework for ranking of cloud computing services. *Future Generation Computer Systems, 29*(4), 1012-1023. https://doi.org/https://doi.org/10.1016/j.future.2012.06.006

Govindarajan, K., Meng, K. C., Ong, H., Tat, W. M., Sivanand, S., & Leong, L. S. (2014). Realizing the quality of service (QoS) in software-defined networking (SDN) based cloud infrastructure. Information and Communication Technology (ICoICT), 2014 2nd International Conference on.

Guck, J. W., Reisslein, M., & Kellerer, W. (2015). Model-based control plane for fast routing in industrial QoS network. 2015 IEEE 23rd International Symposium on Quality of Service (IWQoS).

Hammadi, A. M., & Hussain, O. (2012). A framework for SLA assurance in cloud computing. Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on.

Hongyu, P., Weidong, W., Chaowei, W., Gang, C., & Yinghai, Z. (2015). QoS-guaranteed energy saving routing strategy using SDN central control for backbone networks. *The Journal of China Universities of Posts and Telecommunications, 22*(5), 92-100.

Hu, C., Wang, Q., & Dai, X. (2015). SDN over IP: enabling internet to provide better QoS guarantee. 2015 Ninth International Conference on Frontier of Computer Science and Technology (FCST).

Hussain, W., Hussain, F. K., Hussain, O., Bagia, R., & Chang, E. (2018). Risk-based framework for SLA violation abatement from the cloud service provider's perspective. *The Computer Journal, 61*(9), 1306-1322.

Hussain, W., Hussain, F. K., Hussain, O. K., Damiani, E., & Chang, E. (2017). Formulating and managing viable SLAs in cloud computing from a small to medium service provider's viewpoint: A state-of-the-art review. *Information Systems, 71*, 240-259. https://doi.org/https://doi.org/10.1016/j.is.2017.08.007

Ishimori, A., Farias, F., Cerqueira, E., & Abelém, A. (2013). Control of multiple packet schedulers for improving QoS on OpenFlow/SDN networking. Software Defined Networks (EWSDN), 2013 Second European Workshop on.

Jadhav, B., Saquib, Z., & Pawar, S. (2017). Issues and parameters for improving QoS and performance in SDN. *International Journal of Advances in Electronics and Computer Science, ISSN*, 2393-2835.

Jeong, K., Kim, J., & Kim, Y.-T. (2012). QoS-aware network operating system for software defined networking with generalized OpenFlows. Network Operations and Management Symposium (NOMS), 2012 IEEE.

Karakus, M., & Durresi, A. (2017). Quality of service (qos) in software defined networking (sdn): A survey. *Journal of Network and Computer Applications, 80*, 200-218.

Karaman, M. A., Gorkemli, B., Tatlicioglu, S., Komurcuoglu, M., & Karakaya, O. (2015). Quality of service control and resource prioritization with Software Defined Networking. Network Softwarization (NetSoft), 2015 1st IEEE Conference on.

Kassler, A., Skorin-Kapov, L., Dobrijevic, O., Matijasevic, M., & Dely, P. (2012). Towards QoE-driven multimedia service negotiation and path optimization with software defined networking. Software, Telecommunications and Computer Networks (SoftCOM), 2012 20th International Conference on.

Khan, S. (2015). *Software Defined Network for Energy Efficient Data Centre Operations* [Masters Report].

Khan, S., & Hussain, F. K. (2019). A SOA Based SLA Negotiation and Formulation Architecture for Personalized Service Delivery in SDN. International Conference on Network-Based Information Systems.

Khan, S., Hussain, F. K., & Hussain, O. K. (2021). Guaranteeing end-to-end QoS provisioning in SOA based SDN architecture: A survey and Open Issues. *Future Generation Computer Systems, 119*, 176-187.

Körner, M., Stanik, A., & Kao, O. (2014). Applying QoS in Software Defined Networks by Using WS-Agreement. Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on.

Li, F., Cao, J., Wang, X., & Sun, Y. (2017). A QoS Guaranteed Technique for Cloud Applications Based on Software Defined Networking. *IEEE Access, 5*, 21229-21241.

M. Priyanka, K. J. S. (2014). An Idea for Improvising the Efficiency of SDN Based Business Design with SOA [Joirnal Articla]. *International Journal of Engineering Science and Innovative Technology (IJESIT), 3*(3).

Machado, C. C., Granville, L. Z., Schaeffer-Filho, A., & Wickboldt, J. A. (2014). Towards SLA policy refinement for QoS management in software-defined networking. Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on.

Martini, B., & Paganelli, F. (2016). A service-oriented approach for dynamic chaining of virtual network functions over multi-provider software-defined networks. *Future Internet, 8*(2), 24.

Nawaz, F., Asadabadi, M. R., Janjua, N. K., Hussain, O. K., Chang, E., & Saberi, M. (2018). An MCDM method for cloud service selection using a Markov chain and the best-worst method. *Knowledge-Based Systems, 159*, 120-131. https://doi.org/https://doi.org/10.1016/j.knosys.2018.06.010

Papneja, R. (2022, 24/09/2018). *5G brings a world of new opportunities to service providers. However, these benefits are not realized if networks are not up to the task at hand*. Prodapt Chase Extraordinary. Retrieved 14/04/2022, from https://www.prodaptconsulting.nl/wp-content/uploads/2012/11/Prodapt-Consulting-White-paper-SDN-NFV-2015-V1_1-EN.pdf

Porxas, A. X., Lin, S.-C., & Luo, M. (2015). QoS-aware virtualization-enabled routing in software-defined networks. 2015 IEEE International Conference on Communications (ICC).

Rehman, Z.-u., Hussain, O. K., & Hussain, F. K. (2015). User-side cloud service management: State-of-the-art and future directions. *Journal of Network and Computer Applications, 55*, 108-122. https://doi.org/https://doi.org/10.1016/j.jnca.2015.05.007

Rehman, Z. u., Hussain, O. K., Chang, E., & Dillon, T. (2015). Decision-making framework for user-based inter-cloud service migration. *Electronic Commerce Research and Applications, 14*(6), 523-531. https://doi.org/https://doi.org/10.1016/j.elerap.2015.08.002

Rehman, Z. u., Hussain, O. K., & Hussain, F. K. (2012, 9-11 Sept. 2012). Iaas Cloud Selection using MCDM Methods. 2012 IEEE Ninth International Conference on e-Business Engineering.

Seliuchenko, M., Lavriv, O., Panchenko, O., & Pashkevych, V. (2016). Enhanced multi-commodity flow model for QoS-aware routing in SDN. Radio Electronics & Info Communications (UkrMiCo), 2016 International Conference.

Sharma, S., Staessens, D., Colle, D., Palma, D., Goncalves, J., Figueiredo, R., Morris, D., Pickavet, M., & Demeester, P. (2014). Implementing quality of service for the software defined networking enabled future internet. Software Defined Networks (EWSDN), 2014 Third European Workshop on.

Sharma, S., Staessens, D., Colle, D., Palma, D., Goncalves, J., Pickavet, M., Cordeiro, L., & Demeester, P. (2014). Demonstrating resilient quality of service in Software Defined Networking. Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on.

Shuraia Khan, F. K. H. (2019). *A SOA Based SLA Negotiation and Formulation Architecture for Personalized Service Delivery in SDN* [Unpublished manuscript]. University of Technology Sydney.

Tomovic, S., Prasad, N., & Radusinovic, I. (2014). SDN control framework for QoS provisioning. Telecommunications Forum Telfor (TELFOR), 2014 22nd.

Youssef, A. E. (2020). An Integrated MCDM Approach for Cloud Service Selection Based on TOPSIS and BWM. *IEEE access, 8*, 71851-71865. https://doi.org/10.1109/ACCESS.2020.2987111

Chapter 3

PROBLEM DEFINITION

3.1 Introduction

The first chapter highlighted the importance of personalising quality of service (QoS) delivery in software-defined networking (SDN) to maintain a trusting relationship in a service-oriented environment (SOE). The previous chapter reviewed the existing literature on personalised QoS delivery in SDN. Various researchers have made significant contributions to the delivery of QoS in SDN. Of these approaches, controller design-based QoS management, dynamic resource allocation for QoS guarantees, queue scheduling-based QoS management, and the optimal routing approach for QoS guarantees are the most well-known approaches to achieving QoS in SDN. Also, the approaches mentioned by (Machado et al., 2014), (Bhattacharya & Das, 2013; Körner et al., 2014) identify that a service-level agreement (SLA)-based approach needs to be considered to deliver QoS in service-oriented architecture (SOA)-based SDN, as discussed in the literature review (Chapter 2).

Furthermore, the literature review in Chapter 2 reveals that none of the existing proposals offers a complete methodology to assist in constructing and managing the SLA to deliver QoS, maintain a trust relationship in SDN, and enable personalised service delivery in an SOE. Additionally, in the previous chapter, we identified six weaknesses in the existing literature that need to be addressed in order to propose a complete methodology for personalised QoS delivery in SDN. Section 3.3 of this chapter formally explains the problem we address in this thesis. In Section 3.2, we suggest a set of definitions of the terminologies used while defining the problem in Section 3.3. We divide the research issue into six cohesive research issues and define each of these research issues formally in Section 3.4. Section 3.5 outlines the solution proposal and choice of research method for solving the identified research issues. Finally, the chapter is concluded in Section 3.6.

3.2 Key Concepts

In this section, we represent a formal definition of the terms and concepts used to introduce, explain, and formally define the issue addressed in this thesis.

3.2.1 Trust Maintenance/Preservation

We define *trust maintenance or trust preservation* as the interval from the stage at which the positive trust relationship is established to the stage at which trustworthiness values stay at a level comparable to negative trust (Fachrunnisa, 2011). Fachrunnisa (2011) identifies that the trust-based relationship follows an evolutionary pattern that consists of three stages: building, maintaining, and declining. The state of the trust level may change over to the trust maintenance stage once a positive trust level is established in the trust-building stage.

3.2.2 Hard Trust

Terminology, *Hard trust* can be defined as the trust that represents some related information that is obtained from various security mechanisms. Some examples of security mechanisms are determined by Fachrunnisa (2011) as identity keys, credentials, and certificates.

3.2.3 Soft Trust

Soft trust can be defined as trust information implied from existing experience and observations from others. Soft trust is established in social factors that associate the behaviour with evidence. This association is attained through mapping from observed behavioural evidence (Fachrunnisa, 2011).

3.2.4 Trust-based Interaction or Relationship

A *trust-based interaction* or relationship can be defined as the setting between two parties or agents with the intention of achieving certain pre-defined objectives or goals based on their

trust degree (Chang et al., 2006). This interaction may be attached to either monetary or non-monetary values.

3.2.5 Time-Space of Interaction or Relationship

The terminology *time-space* can be defined as the total duration of interaction time during which the behaviour of the trusted agent will be analysed and the trustworthiness assessment will be carried out (Fachrunnisa, 2011).

3.2.6 Third-party Agent

We define a *third-party agent* as an agent or a party who remains in the middle and in a neutral position in an interaction or settlement. Thus, the third-party agent monitors the interaction of both parties and provides a real-time response in the instance of performance discrepancies (Fachrunnisa, 2011). This third-party agent is considered an impartial, independent, and unbiased body or entity that monitors the trust-based interaction and, in some cases, evaluates the performances in a neutral position between both parties.

3.2.7 Online Monitoring

We define *online monitoring* as monitoring an agreement that needs periodic testing to determine whether all relevant parties have met the agreement terms. In this type of monitoring, the monitoring interval must be specified appropriately, such as daily or hourly, depending on the duration of the agreement and the nature of the agreement terms.

3.2.8 Proactive Continuous Monitoring

The terminology, *proactive continuous monitoring* can be defined as the real-time performance monitoring of how the service provider delivers the service to the requester or consumer as agreed and vice versa (Fachrunnisa, 2011).

3.2.9 Passive Performance Monitoring

Passive monitoring can be defined as an offline monitoring scheme that uses cryptographic primitives to prove that specified checkpoints in the enactment have been reached correctly (Quillinan et al., 2010).

3.2.10 Service

Service is a compilation of a predetermined number of non-traversing and conceivably cohesive sets of activities operated by a service provider to achieve the desired results for a service requester or consumer (Chang et al., 2006).

A service can represent a predictable number of clearly defined activities. Additionally, the activities that compose the service are mutually exclusive because they do not overlap.

3.2.11 Service Requester or Service Consumer

A *service requestor or service consumer* can be defined as an agent or party who has requested a given service from another agent with financial or economic value.

3.2.12 Service Provider

A *service provider* is an agent providing a service with financial or economic value to the service requestor.

3.2.13 Service-Oriented Environment

The *service-oriented environment (SOE)* is a cooperative, shared, and accessible community orientation so that any authorised agent or party can utilise its infrastructure and technology to carry out the business activity(ies) (Hussain, 2006). An SOE comprises various components such as agents, business activities, infrastructure, technology, intermediate agents, and services.

3.2.14 Quality of Service

Quality of service (QoS) is one of the most well-known terms in SOE. QoS is defined as a measurable quantity or measurable unit that implies the worth or value of the services delivered by the service provider to the service requester (Fachrunnisa, 2011).

3.2.15 Interaction or Transaction

We define an *interaction* or *transaction* as communication and settlement between two entities to achieve certain pre-defined objectives or goals (Fachrunnisa, 2011).

3.2.16 Reputation-Based Trust Decision Making

We consider *reputation-based decision-making* or *reputation-based trust development* as reputation data-driven decision-making in which one agent utilises only the experience(s) of other agents to assist in informed decision-making regarding a decision to interact or to make any commitment.

3.2.17 Formulation of Service Requirements

The formulation of service requirements is the process of determining, articulating, and prioritising the service requirements with a quantitative, qualitative, or hybrid expression. The service requirements can be expressed as numeric, text or a combination of numeric and non-numeric measures (Fachrunnisa, 2011) and play a pivotal role in contributing to successful SLA management and creating a trusting relationship between service providers and consumers.

3.3 Problem Overview and Problem Definition

In the literature review (Chapter 2), we argued that delivering personalised QoS in SDN in an SOE is critical in maintaining a trusting relationship; therefore, it requires more attention from all interacting parties. The existing research evidenced the importance of trust maintenance. It can be summarised as an evolutionary pattern and requires more attention after establishing the trust level. Trust is also fragile in virtual environments such as SOEs (Jarvenpaa et al., 1998; Jarvenpaa & Leidner, 1999; Meyerson et al., 1996); thus, a relentless effort to maintain it is essential. Moreover, developing a trusting relationship requires much effort and cost from both interested parties. Therefore, when it is well established, both parties must adhere to a preservation plan; otherwise, the relationship descends into disrepair (Currall & Epstein, 2003). Jarvenpaa (1998) argue that the perception of trust maintenance is used interchangeably or synonymously with the perception of building trust or trust development. This definition of perception leads to confusion and differs from trust-building and trust maintenance and the methodologies/activities implemented to achieve it.

On the other hand, the terms trust and reputation have an integral connection and have been explained in various practices by different researchers. (Hussain, 2006) stated that it is imperative to define trust and reputation in terms of their context-specific and dynamic nature. Otherwise, the trust value-based decision-making process and the reputation-based decision-making process will not take into account these crucial aspects. In the literature (Hussain, 2006), trust focuses on defining trust and reputation in a generic sense to apply to all domains. Furthermore, (Damiani et al., 2002; Damiani et al., 2003) propose a solution that assists in reputation-based decision-making and provides a secure means for communicating

recommendations. Damiani et al.(2002) define trust and reputation in the peer-to-peer environment, grid computing and the client-server domain. (Hussain, 2006) defines trust and reputation in SOEs as trust and reputation in this environment have unique features that cannot necessarily be applied to other domains. Therefore, reputation-driven decision-making is one of the critical areas in service-oriented-based technologies such as SDN, which requires further research.

In addition, the literature review (Chapter 2) illustrates that the work on guaranteeing end-to-end QoS provisioning in SOA-based SDNs focuses mainly on identifying various approaches to achieve QoS, which is not specific to SDN but also to a generic SOE. One of the significant challenges for SDN providers is to ensure users' satisfaction. Existing approaches have attempted to address this issue. The SLA-driven approach and most queue scheduling approaches have provided QoS solutions for SDN networks rather than application or area-specific solutions. Moreover, very little research has been done to affirm the successful response rate after implementing their approaches. Most research has been tested in simulation, emulation, or test bed environments and not on the entire SDN network.

Furthermore, no approaches can provide reliable personalised service delivery for service provider selection and service request acceptance decision-making based on the comparative analysis. Therefore, delivering reliable personalised services to the SDN consumer remains an unresolved gap in the research area of SDN. Moreover, no unique measurement technique in SDN can authenticate the customer's satisfaction level with the provided services. As a result, there is always a gap between the service provider and consumer in relation to service satisfaction information that can be considered for service provider selection decision–making which generates a lack of transparency between service providers and consumers. Furthermore, none of the previous research proposes an SOA-based QoS delivery in SDN except for the work in (Bueno et al., 2013; Duan, 2014). As a result, there is scant research on personalised service selection and guaranteed QoS delivery of SDN in SOA-based architecture and achieving QoS in SOA-based SDN has not been investigated. Moreover, to the best of our knowledge, the existing literature does not provide a complete methodology that assists in maintaining or preserving a trusting relationship between the service provider and consumer.

We have argued that an SOA-based SDN needs to provide a network that satisfies the requirements of different services that must follow some similar approach (proposed in the existing research) that first ensures the correct identification of services according to the needs of the applications, forms SLAs with them and manages them to ensure that expectations are achieved. We have also argued in the literature review (chapter 2) that four essential requirements (R1, R2, R3, and R4) have been defined and discussed to achieve such an approach to guarantee end-to-end QoS provisioning in SOA-based SDN architecture. Given the importance of guaranteed reliable and personalised service delivery in SDN to maintain trust, there is the need for a complete methodology in a service oriented SDN environment. We also stated that using the SLA-based QoS delivery methodology for maintaining trust refers to preserving the existing trust level condition or increasing it to a higher level. The methodology encompasses four frameworks that can be used to achieve the objective of a personalised QoS guarantee in SDN.

In the existing study on the SOA-based framework, one of the strategies to maintain a coherent relationship between all interacting parties is to involve a neutral party in a relationship. Several studies define the term hard trust relationship or security platform where the third-party or intermediate agent is involved in the interaction to preserve the trust level. However, no investigation has been conducted to understand the influence of third-party agents on another type of trust relationship named soft trust. In contrast, other studies (Greenberg et al., 2007; Jarvenpaa & Leidner, 1999) conclude that one effective way to ensure a trusting relationship can be preserved or maintained is by involving an intermediary or third-party agent. An intermediate agent or third party can be defined as an agent that connects the service provider and consumer to achieve a contract or reconciliation.

An (SLA-oriented approach is suggested in several existing studies to deliver QoS in an SOE. It is also suggested that contract conformance or formulating an SLA is an effective approach to preserving the trust relationship in a service-based transaction. The research also noted that maintaining a trusting relationship in a service-based transaction plays a significant role in maintaining service quality and timely delivery. With contract conformance, both parties need to confirm whether they have studied all the provisions in the business agreement. Contract conformance is an agreement between two parties as to how they will create a collaborative

relationship for a period of time in terms of service request delivery. However, in the existing literature, no framework is proposed for formulating and negotiating the service requirements in attaining this agreement. Furthermore, to the best of our knowledge, no framework is proposed in the existing literature that considers establishing an SLA in order to maintain trust maintenance relationships. The existing literature extensively discusses the need to negotiate and formulate an SLA during the trust-building stage, but very limited proposals discuss an approach to formulating the service requirements.

Another essential component in the existing literature is SLA control and performance monitoring to maintain trust relationships (Das & Teng, 1998; Jagers et al., 1998; Kusari et al., 2005). Various studies have suggested that the QoS of the delivered services should be monitored constantly to ensure the SLA is well maintained, as there is an internal relation between the QoS delivery and maintaining trust. However, the literature does not provide any QoS monitoring and SLA monitoring strategy to ensure that the trust relationship is maintained or that there is a continuity of trust. Control and monitoring always occur in the completion of service delivery at the end of a transaction to determine the quality of the performance and sustainability of the SLA. Hence, we need a service monitoring or observation framework to maintain the performance quality that is agreed upon in the SLA (and ensure the performance discrepancy does not exceed a pre-defined threshold). This observation framework enables monitoring or detecting changes in the service quality performance level in a real-time and continuous manner and immediate response to such changes. Such a method for monitoring actual performance progress against agreed performance should be visible to all interacting parties so that any early discrepancies can be detected and resolved proactively. Moreover, in the existing literature, no effort has been made to identify any network openness or possible areas that may cause service discrepancies. In addition, there is no developed framework by which both parties can resolve the performance discrepancies without prolonging the service delivery time as agreed.

The basic assumption which can be made from reviewing the existing literature on SLA preservation and maintaining a trust relationship is that both the service consumer and provider reach a certain trust level through previous interactions that occurred during the service negotiation and conformance stage. The success of sustaining the SLA can be determined if

the deliverability of service performance quality at the end of the time-space of interaction is greater than or equal to the condition of the SLA at the beginning (Fachrunnisa, 2011). The existing research shows that no framework evaluates the service quality performance against the quality of the performance committed to in the SLA. Hence, to determine the success of the SLA or to preserve the SLA, we need a service evaluation framework to identify any service discrepancies that occur during the interaction. The framework delivers an accurate and unbiased evaluation outcome that provides transparency between all interacting parties and is open to other options.

Moreover, after assessing the service performance at some point during the interaction, there is a need for intelligent decision-making to look for options based on the assessment results. No approach in the existing literature can utilise intelligent decision-making in service continuity. After evaluating the performance, there may be several decision options, such as modifying the SLA or renegotiating the SLA. Hence, we need a framework that assists us in determining the service continuity decision-making that will help to open options for renegotiation or to repair the trust relationship between all interacting parties.

The above problem that described leads to our proposal of a complete methodology for personalising QoS delivery in SOA-based SDN and maintaining the trust relationship that takes into account the following frameworks: formalisation and negotiation of service requirements, service management framework that includes passive and proactive continuous performance monitoring, service performance evaluation, and service continuity intelligent decision-making framework. Hence, we formally define the problem we intend to address in this thesis. Thus, we proposed the following research question and sub-questions:

In a service-oriented environment, how can we provide a reliable approach to ensure guaranteed and personalised service management for informed decision-making in software-defined networks?

We break the research question into the following five research sub-questions:

> Sub-Question 1: *How can we develop an intelligent method for personalised service delivery in SDN?*

Sub-Question 2: *How can we develop an intelligent method for SDN service monitoring and proactive violation prediction?*

Sub-Question 3: *How can we guarantee reliable service delivery in SDN?*

Sub-Question 4: *How can we develop intelligent decision-making for service continuity?*

Sub-Question 5: *How can the proposed methods be validated and verified?*

3.4 Research Issues

This section discusses the research issues that need to be addressed to solve the aforementioned problem concerning personalised service delivery and ensuring QoS in a service-oriented SDN. The research issues that need to be addressed are as follows:

1. In the service-oriented SDN context, we define the following terms: service-oriented SDN, personalised service delivery, QoS delivery, reputation data, reputation ranking, SLA, mutually agreed behaviours and trust maintenance.

2. We propose a methodology for service negotiation and to formulate the service requirements that enable personalised service delivery in SOA-based SDN where two interacting parties agree on an SLA formulation during their interaction to build a trusting relationship. We need (i) a methodology for a service negotiation lifecycle process, (ii) a methodology for the service provider to evaluate the suitability of accepting a request from the service consumer, and (iii) a methodology for service provider selection from a list of interested providers. This process involves various activities, such as articulating the service requirements, negotiating the service requirements, and renegotiating specific service requirements by both parties.

3. We propose a methodology for service management using passive and real-time proactive continuous performance monitoring of service deliverability. We need a methodology by which a third party or an intermediate agent monitors both parties' performance proactively, attains knowledge, and determines the possible service

discrepancies of the high-priority services and actions accordingly. The framework encompasses (i) a methodology of continuous proactive service monitoring to monitor the services in a certain timeframe continuously to identify any risk of service discrepancy; and (ii) a methodology of passive service monitoring to determine the actions required to deliver business requirements.

4. We propose a methodology that measures the QoS to ensure the services which are delivered meet the SLA. The methodology will enable a third party or intermediate agent to neutralise the delivered services and identify any service discrepancies that may lead to a possible service level agreement violation. The methodology will encompass (i) measurement services to analyse the proactive performance monitoring status and arrange the network traffic according to evaluation metrics requirements; (ii) condition evaluation services to evaluate any service discrepancies.

5. We propose a framework for making intelligent service continuity decisions with the same interacting parties. We need a methodology by which a third party or intermediate agent can demonstrate an informed decision to all interacting parties on whether to continue the current agreement or renegotiate at the end of the interaction time.

6. We develop a proof of concept to validate the proposed methodology.

The following section outlines the above research issues that need to be addressed to solve the problem stated in Section 3.4.

3.4.1 Research Issue 1: Conceptual Definition

The first research issue is to define, in the context of an SOA-based SDN environment, the concepts of personalised service delivery, QoS delivery, reputation data, reputation ranking, SLA, mutually agreed behaviour, and trust maintenance, taking into account their context-specific and dynamic nature. As discussed in Chapter 2, service personalisation and delivering QoS have been identified as critical requirements in SOA-based SDN and are defined by researchers in various ways. We have identified that personalised QoS delivery is related to

trust maintenance and enhancement. Trust has been identified as a multi-disciplinary field with applications in varied disciplines (Hussain, 2006) and explained as a trust evaluation with three stages (Fachrunnisa, 2011). The main objective of developing a personalised QoS delivery using negotiation steps to maintain a trust relationship is to represent that trust has a dynamic nature in the context of any interaction. The development of service negotiation from one stage to another stage depends on the effort made by both parties to manage the trust level condition in their mutual relationship.

This thesis proposes a definition for developing personalised QoS in an interaction that leads to a formal agreement between both parties. Chapter 4 proposes a formal definition of personalised service delivery in SOA-based SDN, enhancing trust relationships and their accompanying terms. Furthermore, we describe certain characteristics of these terms and their relationships.

3.4.2 Research Issue 2: Propose a Framework for Service Negotiation and Formulate the Service Requirements

In this thesis, we reach the viewpoint that the service provider requires contract conformance with the service consumer in the interaction. The contract conformance or SLA facilitates both parties' performance monitoring and evaluation of service delivery quality. In Chapter 2, we identified that contract conformance or a business contract is an element that can be employed to maintain trust. However, as shown in the existing literature, the concept of constructing an SLA is not considered a primary objective of trust maintenance. Also, there is no framework by which both interacting parties can formulate their service requirements during the trust maintenance stage.

Additionally, the literature review discusses several researchers who acknowledge QoS delivery in an SOE. We argue that, in order to achieve the maximum benefit from a relationship in SOEs and to ensure QoS, both parties (the service provider and service consumer) need to interact with each other in the presence of a third-party agent/intermediate agent as a neutral party. This intermediate agent plays an important role as mediator/arbitrator/conciliator all

through the relationship and is responsible for responding on a proactive and corrective basis. Moreover, the intermediate agent concentrates on articulating the service requirements agreed upon by both parties, assisting in resolving any conflict that arises during the interaction to maintain trust.

In SOEs, trust is more rational than emotional; preserving a trust relationship can best be accomplished by measuring the level to which a service task has been undertaken. (Hussain, 2006) stated that the trust level can be determined by correlating actual performance and agreement with performance. Thus, in our viewpoint, to sustain a trusting relationship, both parties should consider a benchmarking-based approach that can evaluate their performance when carrying out an interaction. Specifically, in an SOE, it is argued that the actual performance can be seen as the substantial service delivery, while agreed performance is the set of service requirements agreed upon by both parties. Therefore, a formal contract such as an SLA, which is defined as a negotiated agreement between two parties (the service provider and the service requester), specifies the agreed service requirements and is essential for preserving trust.

In contrast, enhancing and preserving the trust relationship between both parties entails articulating the service requirements, negotiating the service requirements, determining and improving the conflicting service requirements, and renegotiating certain service requirements agreed upon by both parties. However, there is no framework in the existing literature that introduces an SLA that is agreed upon between two interacting business parties for trust maintenance in an SOA-based SDN. Therefore, this research finds a primary need for a method for agreement between both parties on the service requirements that is essential for trust building and maintenance. Chapter 4 presents a high-level overview of a framework involving the formulation of service requirements and negotiation to draw up an SLA for interaction in the trust maintenance stage. In Chapter 5, we present the detailed mechanism of the framework.

3.4.3 Research Issue 3: Propose a Framework for Service Management Using Passive and Proactive Continuous Performance Monitoring

Following the research on ensuring the QoS in service delivery in SDN, we conclude that QoS-guaranteed service delivery means providing and ensuring the service contract's required services. Moreover, we recognise that control and monitoring are essential factors considered in the existing research to sustain trust in an SOE. In addition, the existing research argues that there are several approaches to control and monitor service deliverability and suggests that the level of trust in interaction should be continuously monitored to ensure the success of the interaction. Hence, even though trust is formed between the service provider and service consumer during the interaction, service monitoring plays a significant role in ensuring that the business relationship continues to operate as expected (Fachrunnisa, 2011). However, no researchers are able to monitor the runtime network and use the monitoring log that enables proactively providing SLA violation prediction information.

Different to the existing research on control and monitoring to minimise the risk of agreement violation and preserve the trust relationship, in this thesis, we propose a framework of proactive, continuous monitoring during the interaction to maintain the trust degree. Moreover, we propose a framework for passive monitoring to analyse the current network status and determine the vulnerabilities that assist in determining the focus area and actions. The performance is observed at periodic performance assessments at intermediate checkpoints rather than at the end of the interaction. The objective of the framework is to monitor the performance during the course of the transaction to minimise the discrepancy between actual performances and the mutually agreed performance rather than monitoring the performance of the services at the end of the interaction. In Chapter 4, we present an overview of the framework for service management consisting of proactive continuous monitoring and passive monitoring to preserve the trust relationship. Details of this framework are presented in Chapter 6.

3.4.4 Research Issue 4: Propose a framework for Service Evaluation to Identify Any Performance Discrepancy

In service provider and consumer relations, the term reliable indicates that there is a minimal possibility of the agreement being breached during the interaction. Violating an SLA may require significant financial penalties for a service provider. Sometimes, consumer companies may also face significant financial loss and other related losses, such as loss of reputation for service disruption. Hence, service providers and consumer companies mandate reliable service delivery to avoid these consequences. The existing studies detailed in the literature review discuss the QoS guarantee subjectively, which means there is no unified approach that can explicitly measure the QoS guarantee degree (objective-based).

Additionally, there are no approaches that are able to compare the deliverability of the service quality against service quality contract conformance. In this research, we propose an intelligent framework to ensure service delivery reliability in SDN with an agreed quality by evaluating and investigating any performance discrepancy. We present an overview of the framework with associated stages for service evaluation to determine any performance discrepancy. Details of the framework are presented in chapter 7.

3.4.5 Research Issue 5: Propose a Framework for Making an Intelligent Decision Regarding Service Continuity

The research issue we need to address in this thesis to provide a complete methodology for SLA management is a framework for personalised QoS delivery in SOA-based SDN and preserving the trust relationship between all interacting parties. As the objective of trust preservation is to sustain the trust relationship during the interaction, we need a framework for service contract continuity decision-making. This intelligent decision-making needs to consider service management, which consists of proactive and passive monitoring and a condition evaluation mechanism that we discussed in the previous research issue. Furthermore, since both interacting parties have a mutual agreement about the service context and its service criteria at the beginning of the interaction, contract continuity decision-making also needs to

be considered in the mutual agreement, which would serve as the integral component of the SLA management lifecycle framework. Chapter 4 presents an overview of the framework for service contract continuity and intelligent decision-making, and the details of this framework are presented in Chapter 8.

3.4.6 Research Issue 6: Validate the Proposed Methodology by Simulation Experiments

To confirm the completeness of the proposed framework, we need to validate the proposed solutions for the aforementioned research issues. In the validation, we develop an approximation or representation of prototyping systems based on the proposed personalised QoS delivery in SOA-based SDN. To validate the conceptual framework, we prototype all of the frameworks and discuss the results. We present the solution overview for this research issue in Chapter 4 and the prototypes to validate the proposed frameworks in the related chapters.

3.5 Research Approach in Solving Research Issues

To address the research issues, this thesis focuses on developing, examining, and validating a methodology for preserving trust in SOA-based SDN. In order to propose solutions for the six research issues identified in the previous section, a systematic scientific approach needs to be followed to ensure that the methodology development is aligned and scientifically oriented. Therefore, in this section, we overview the existing scientifically oriented research methods and rationalise our selection of a particular research approach.

3.5.1 Research Methods

According to the research study and its nature, we identify that the design science research approach is the most appropriate for our purposes. Design science research is a problem-solving paradigm that seeks to enhance human knowledge by creating innovative artifacts(vom

Brocke et al., 2020). Design science research follows a pattern such as; the development of new techniques, architecture, methodologies, devices, or concepts; therefore, they intermingled together and produced a new theoretical framework(Fachrunnisa, 2011). The design science research method generally detects problems and proposes solutions to address these problems. Several studies (Galliers, 1992; Hevner et al., 2004; Peffers et al., 2007) demonstrated a conceptual framework for design science research that ascertained an understanding of the problem domain and constructed a solution by developing applications or artifacts. This research type involves validating theoretical predictions, and particularly in engineering, the spirit of "making something work" is essential (Galliers, 1992).

This thesis develops a new methodology for personalised QoS delivery in an SOA-based SDN. We design a methodology and validate it using experimental simulation instead of building and evaluating a hypothesis. Hence, our research falls under the design science research approach (Galliers, 1992), the most commonly used research approach for science and engineering-based research, developed by Simon in 1969 (Simon, 1996). His study of artificial intelligence involves the artifact's development to solve a problem and therefore emphasises the practical nature of the constructed artifact (Hevner et al., 2004; March & Smith, 1995). Furthermore, the design science approach incorporates behavioural science and engineering, which leads to the development and justification of a theoretical framework; therefore, it develops and evaluates the artifact features of the research. The approach comprises three levels: conceptual level, development or perceptual level, and impact or validation or practical level (Nunamaker Jr et al., 1990). The details of these levels are described in the following.

3.5.2 Conceptual Level

This is the first level of the design science research method, which primarily identifies the problems and defines the problems formally through a systematic analysis process. The ideas, concepts, and conceptual framework structure are developed in this initial stage. The details of the framework are provided in the next stage.

3.5.3 Perceptual Level

After formulating the identified problems, a conceptual framework is constructed for the proposed methodology; therefore, a tool, system, or environment is developed. In this level, implementations and prototyping is performed. These instantiations are necessary to challenge or support the theory that is constructed in the previous stage.

3.5.4 Practical Level

In the impact or validation or practical level, testing and validation are performed through experimentation with real-world examples in a laboratory or field testing(Fachrunnisa, 2011). The validation of the methodology is required to realise the benefits in both technological and social contexts. This realisation leads to determining the future work scope and improving the proposed methodology. Fine-tuning the methodology is the most critical aspect of this stage.

3.5.5 Choice of Design Science Research Methodology

In this thesis, the design science research methodology is chosen as the research method for the proposed solution development due to the nature of the research. An overview of this research approach is illustrated in Figure 3.1.

In the initial stage, we identified the research issues by collecting and analysing a large body of literature on topics related to the study. According to the findings from the extensive review, we formulated the issues that need to be addressed. This research issue was formally expressed in Section 3.3. Consequently, considering the dynamic nature of interactions, we outlined several fundamental concepts (such as personalised QoS, SOA-based SDN, trust maintenance or trust preservation, SOA principles, etc.). These terms are used during the conceptual solution development stage. In the next stage, we formulated the conceptual solution for the identified issues being addressed in this thesis. We developed the methodology for personalised QoS delivery in SOEs based on SDN at the perceptual level. Subsequently, we engineered prototype

systems and established some case studies that will be used later to test our proposed methodology

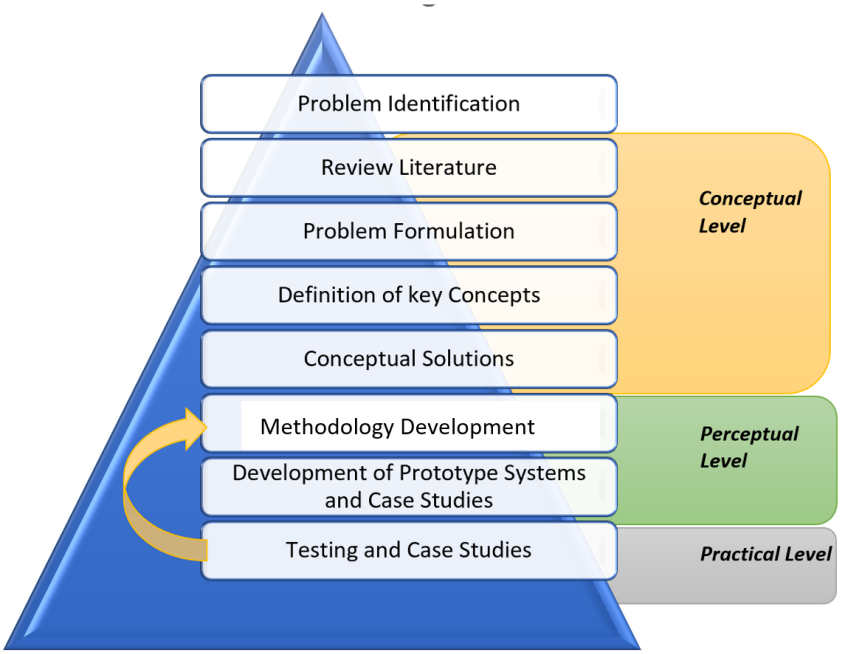


Figure: 3.1 The design science research approach (Nick Toscano, 2011; Simon, 1996; SISAK, 2018)

As shown in Figure 3.1, the methodology development process and prototype development with case studies fall under the perceptual level. Once the prototype systems were engineered, we validated our proposed methodology using it and the developed case studies. Therefore, we evaluated and validated our proposed methodology based on the results obtained at the practical level. Furthermore, we fine-tuned our proposed methodology based on the evaluation and validation results. Hence, the evaluation and validation of the developed methodology fall under the practical level.

3.6 Conclusions

This chapter formally defines the problems we address in this thesis. Consequently, the identified research problem was divided into seven unified research issues. In order to address the problem, we need to resolve the seven issues mentioned in this chapter, and this is the primary motivation for this thesis. Each of the seven identified research issues was explained comprehensively in relation to the existing literature. Furthermore, we summarised the different research approaches and indicated that we anticipate implementing the design science research approach in this research.

In the next chapter, we present an overview of the solution to the problem being addressed in this thesis. In addition, we outlined an overview of the solutions for each of the seven research issues that comprise the problem being addressed in this thesis. The detailed framework of the SLA-based methodology for personalised QoS delivery is then described in Chapters 5-8.

3.7 References

Bhattacharya, B., & Das, D. (2013). SDN based architecture for QoS enabled services across networks with dynamic service level agreement. Advanced Networks and Telecommuncations Systems (ANTS), 2013 IEEE International Conference on.

Bueno, I., Aznar, J. I., Escalona, E., Ferrer, J., & Garcia-Espin, J. A. (2013). An opennaas based sdn framework for dynamic qos control. 2013 IEEE SDN for Future Networks and Services (SDN4FNS).

Chang, E., Hussain, F., & Dillon, T. (2006). *Trust and reputation for service-oriented environments: technologies for building business intelligence and consumer confidence*. John Wiley & Sons.

Currall, S. C., & Epstein, M. J. (2003). The fragility of organizational trust:: Lessons from the rise and fall of Enron. *Organizational Dynamics, 32*(2), 193-206.

Damiani, E., di Vimercati, D. C., Paraboschi, S., Samarati, P., & Violante, F. (2002). A reputation-based approach for choosing reliable resources in peer-to-peer networks. Proceedings of the 9th ACM conference on Computer and communications security.

Damiani, E., Di Vimercati, S. D. C., Paraboschi, S., & Samarati, P. (2003). Managing and sharing servants' reputations in P2P systems. *IEEE Transactions on Knowledge and Data Engineering, 15*(4), 840-854.

Das, T. K., & Teng, B.-S. (1998). Between trust and control: Developing confidence in partner cooperation in alliances. *Academy of management review, 23*(3), 491-512.

Duan, Q. (2014). Network-as-a-service in software-defined networks for end-to-end QoS provisioning. 2014 23rd Wireless and Optical Communication Conference (WOCC).

Fachrunnisa, O. (2011). *A methodology for maintaining trust in virtual environments*, Curtin University].

Galliers, R. (1992). *Information systems research: Issues, methods and practical guidelines*. Blackwell Scientific.

Greenberg, P. S., Greenberg, R. H., & Antonucci, Y. L. (2007). Creating and sustaining trust in virtual teams. *Business horizons, 50*(4), 325-333.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75-105.

Hussain, F. K. (2006). *A Methodology for Trust Management in Service Orientated*

Environment, Curtin University of Technology].

Jagers, H., Jansen, W., & Steenbakkers, G. (1998). Trust and Control in Virtual Organizations. *PrimaVera Working Paper, 98*.

Jarvenpaa, S. L., Knoll, K., & Leidner, D. E. (1998). Is anybody out there? Antecedents of trust in global virtual teams. *Journal of management information systems, 14*(4), 29-64.

Jarvenpaa, S. L., & Leidner, D. E. (1999). Communication and trust in global virtual teams. *Organization science, 10*(6), 791-815.

Körner, M., Stanik, A., & Kao, O. (2014). Applying QoS in Software Defined Networks by Using WS-Agreement. Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on.

Kusari, S., Cohen, D., Singh, J., & Marinova, D. (2005). TRUST AND CONTROL MECHANISMS IN ORGANIZATIONAL BOUNDARY SPANNERS'COGNITIONS AND BEHAVIORS. Academy of Management Proceedings.

Machado, C. C., Granville, L. Z., Schaeffer-Filho, A., & Wickboldt, J. A. (2014). Towards SLA policy refinement for QoS management in software-defined networking. Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision support systems, 15*(4), 251-266.

Meyerson, D., Weick, K. E., & Kramer, R. M. (1996). Swift trust and temporary groups. *Trust in organizations: Frontiers of theory and research, 166*, 195.

Nick Toscano, P. M. (2011). *Rio Tinto still facing 'long road ahead' after cave blast disaster*. https://www.smh.com.au/business/companies/rio-tinto-still-facing-long-road-ahead-after-cave-blast-disaster-20210930-p58vzw.html

Nunamaker Jr, J. F., Chen, M., & Purdin, T. D. (1990). Systems development in information systems research. *Journal of management information systems, 7*(3), 89-106.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems, 24*(3), 45-77.

Quillinan, T. B., Clark, K. P., Warnier, M., Brazier, F. M., & Rana, O. (2010). Negotiation and monitoring of service level agreements. In *Grids and Service-Oriented Architectures for Service Level Agreements* (pp. 167-176). Springer.

Simon, H. (1996). The Sciences of the Artificial 3rd Edition Massachusetts. MIT.

SISAK, T. (2018). *The Pilbara*. https://scoop.com.au/guide-to-the-pilbara/

vom Brocke, J., Hevner, A., & Maedche, A. (2020). Introduction to design science research. In *Design science research. Cases* (pp. 1-13). Springer.

Chapter 4

SOLUTION OVERVIEW

4.1 Introduction

Personalised and reliable service delivery is crucial in ensuring QoS in SOA-based SDN. Several existing research works have proposed and developed components/mechanisms/frameworks, as detailed in our literature review (Chapter 2). Of these approaches, very little research focuses on ascertaining the reliability and trust relationship maintenance in a service-oriented environment. On the other hand, personalisation is an advanced feature in an SOA that engenders reliability and improves the trust between the service provider and consumer (Chang et al., 2006; Fachrunnisa, 2011; Hussain, 2006). There is no existing research that studies personalised service delivery in SOA-based SDN. From the discussion in Chapters 2 and 3, it is evident that none of the existing studies provides a complete methodology for preserving a trusting relationship between a service consumer and a service provider in terms of delivering a guaranteed, personalised, and reliable service in SOA-based SDN. Chapter 3 identified six research issues that must be addressed to solve this pivotal problem.

This chapter presents an overview of the solutions to the six research issues. We discuss the three stages of the trust evaluation model: trust-building, trust maintenance, and trust decline in terms of a service-oriented business model in Section 4.2.1, Section 4.2.2, and Section 4.2.3, respectively. This thesis focuses on the trust development stage, followed by a comprehensive focus on the trust maintenance stage; therefore, in Section 4.3, we propose an SLA-based solution for trust maintenance in service-oriented SDN. Subsequently, we present the solution to each research issue identified in the problem overview chapter (Chapter 3), followed by the chapter conclusion.

4.2 Overview of the Solution for the Definition of the Trust Development Lifecycle in a Service-Oriented Environment

Trust evaluation in a (long-term) relationship in a service-based environment is the foundation for building a successful relationship between the service provider and the consumer. As

discussed in Chapter 2 and Chapter 3, the existing literature has a minimal focus on trust development and evaluation in a service-oriented environment. From this identified gap, we are motivated to develop a comprehensive understanding of the importance of trust relationships in a service-oriented environment and the co-relation between the evaluation of trust with relationship development in SOA.

4.2.1 Trust in the Business Context

To describe trust in the business context, first, we define the term trust. Trust is considered a primary ingredient in any relationship, especially in places where transactions are carried out in an anonymous, pseudo-anonymous, or non-anonymous distributed environment to provide the agreed quality of service (QoS)(Chang et al., 2006)

Similarly, trust is defined by (Hussain, 2006) as the belief that a trusting agent has in the trusted entity's capability and willingness to deliver a mutually agreed service in a given context at a given timeslot. In the following section, we discuss the importance of the trust relationship in service-based architecture, trust in a business context, and the relationship between trust evaluation and relationship development in an SOA-based environment.

In the business context, trust can be defined as the confidence or a feeling of certainty that one person develops in another person or thing with which they are interacting (Chang et al., 2006). Assurance, certainty, and confidence are the primary elements that everyone or every organisation seeks in terms of a reasonable assurance of what they carry out and what they will receive. Thus, trust ensures honest dealings regarding product or service quality typically associated with mutual agreements and beliefs.

The key motivation for developing or building trust is that positive trust assists in building a business reputation that significantly impacts consumer confidence, fair trading, and mutual relationships.

4.2.2 Trust in the Service-Oriented Environment

A service-oriented environment is a collaborative environment that can be described as a shared and open community where agents develop a technology infrastructure to perform business activities. The activities include product sales, service delivery, and information retrieval. The service-oriented environment consists of four elements (Chang et al., 2006): agents, business activities, infrastructure, and technology.

Unlike traditional business perceptions, a service-oriented environment is one where operations are carried out collaboratively, for example, in a web service environment where online users are often anonymous and primarily endorse themselves by posting and answering questions on the web. In such an environment, online users continue to operate through business, research, or industrial collaborations(Chang et al., 2006). Hence, the service-oriented environment operates in a shared environment where the agents share information through the web; almost certainly, the information available is distributed from unknown agents, products, service providers, or merchants. The environment is an open community environment where the agents have the freedom to attach or detach themselves; therefore, a list of pre-identified and authenticated agents must be available as community servers.

Business in a service-oriented environment refers to the business activities, including the management process and workflows that assist a provider company in selling or delivering their products or services using new technologies and infrastructure to maximise consumer confidence and business value (Chang et al., 2006). The discussion above shows that the business operates in an unstructured community environment in a service-oriented environment. Furthermore, the current trend is moving from small closed communities involving direct interaction and static binding to more open, indirect, and dynamic interaction (Chang et al., 2006). Trust and QoS constitute significant concerns in such an environment. Introducing trust and trustworthiness in technology, including recommendation and rating systems, provides some control over such an environment and ensures QoS delivery. These rating systems ensure QoS, give buyers accumulated information about the provider, and

provide some degree of reliability and assurance. Therefore, trust and QoS are vital contributors to the success of a service-oriented environment-based business.

4.2.3 Trust Evaluation Model

The trust evaluation model detailed by Fachrunnisa (2011) comprises three stages: trust building, maintenance, and decline. Our proposed SLA-based personalised QoS delivery framework develops solutions for all three stages of the trust evaluation model. Details of the frameworks are described in Chapters 5, 6, 7, and 8. The following section describes three stages of the trust evaluation model.

4.2.3.1 Trust Building Stage

The trust-building stage is the first stage of the trust evaluation model, where the interacting parties anticipate entering a new relationship. During this stage, the interacting parties ascertain their needs and willingness to form a relationship, leading to a decision to enter into a formal Agreement (Fachrunnisa, 2011). In this stage, building trust between the interacting parties is extremely important. It is also expected that over time and with additional consequent interaction, each other's willingness and ability to interact productively is communicated, clarified, and verified (Fachrunnisa, 2011).

During this stage, the trust level may vary according to the way the trust relationship is built. Thus, the trust level increases or decreases depending on how an agent tries to build their trust level. Thus, trustworthiness may expand either positively or negatively strengthened. On the other hand, the trust-building decision may be unilateral, emanating from one party only (Fachrunnisa, 2011). For example, a service provider may create interest in products or deliver a range of product/service package options with the intention of developing trust with the consumer. However, the service requester may not know that the service provider is attempting to initiate a relationship.

Similarly, the service requester may try to build trust in the service provider by demonstrating that they have a reliable payment performance and making repeat purchases without the service provider being aware of this. Therefore, both the service provider and the service consumer are interested in developing trust due to their interaction. The motivation of this interaction is an object-related exchange or transactional purpose only with unpredictable behaviour(Fachrunnisa, 2011). Therefore, it can be viewed that this level of the relationship is based on transaction or exchange only (Fachrunnisa, 2011). The definition of exchange in terms of transactional behaviours refers to giving and receiving something of value. The relationship between the two interacting agents and the trust level ascends to the second stage if both parties realise there is value in maintaining the relationship. The second stage is characterised by a mutual positive trust level and a certain number of successful interactions being experienced.

From the above discussion, it can be summarised that the trust-building stage traverses the period from when the relationship was initiated to the stage when a positive trust relationship is established. In this stage, the primary focus of both parties (the service consumer or trusting agent and the service provider or trusted agent) is to construct the trust level and relationship value.

4.2.3.2 Trust Maintenance Stage

The trust-building stage generates a certain level of confidence in both parties to progress their relationship to the next level through relationship experience and value discovery. Both parties realise the benefits of forming and sustaining their business relationship. Furthermore, both parties have the intention to establish a positive trust level that needs to be maintained or even increased to a higher level (Hussain, 2006).

The trust maintenance stage is the second stage of the relationship evaluation model, where trust moves from an exchange related to a collaborative business relationship. Both parties engage in effective cooperation rather than an exchange-only relationship to preserve their trust-based business relationship. Therefore, trust needs to be maintained in this stage to sustain

a long-term collaborative business relationship. Both parties must desire to sustain their trusting relationship, as trust maintenance is futile without the willingness of both parties to sustain this relationship. For example, a service provider might expend much energy on maintaining the relationship with the customer in the trust-building stage; however, the customer may not be as committed to the relationship and may be looking for another service provider, which may result in the service provider experience significant loss (Fachrunnisa, 2011).

Maintaining a mutual relationship allows the parties to negotiate with each other if any unexpected event occurs during the interaction. The involvement of a third-party agent who can act as an independent and impartial representative in a service-based business relationship can positively impact this relationship by helping to stabilise the partnership should any unexpected events occur (Fachrunnisa, 2011; Hussain, 2006).

Thus, in this thesis, we explain the trust maintenance stage as; the stage which extends from the moment when positive trust is established to the moment when trustworthiness values are degrading to a level that is comparable to negative trust. Therefore, the common objective of both parties (the service consumer and service provider) is to maintain an acceptable degree of trust in their relationship.

4.2.3.3 Trust Declining Stage

An unsatisfactory or disappointing experience on the part of one or both parties leads to the declining trust stage, where there is no room for further negotiation or re-calibration. Several occurrences may bring about this stage, such as if a service consumer's demand frequently changes or increases to such a high level that the service provider cannot meet their demands, which places the relationship in a vulnerable position. In terms of the trust relationship, this stage is usually triggered when the positive trust relationship cannot be sustained, and neither party perceives any benefits in persisting with the relationship. In this instance, a decision to terminate the relationship could be made by either party. The relationship termination could be unplanned, or, with the help of a third-party agent, the relationship could be under negotiation

in an attempt to mend the relationship. To sum up, we can define the trust declining stage as starting from the period when negative trust has been established up to the period when the trust relationship no longer exists.

In this chapter, we differentiate the three stages of the trust evaluation model and examine the characteristics of each stage. Moreover, this research focuses on the trust-building and trust-maintenance stage s. In the next section, we present an overview of the methodology that assists in building and maintaining trust.

4.3 Overview of the Solution of the Service-Level Agreement (SLA) framework for Guaranteed QoS delivery in SOA-based SDN

To address the research issues identified in the previous chapter, we need to develop a reliable framework to provide a guaranteed QoS in SDN. The challenges discussed in Chapter 3 will be taken into consideration to develop the framework. The high-level design of the SLA-based QoS framework is provided below.

This section provides a detailed overview of the proposed methodology, an SLA-based framework in SDN that can provide personalised service delivery with QoS-guaranteed services for the consumer to preserve the trust relationship. This methodology does not focus on developing trust relationships between the new interacting parties but on how the interacting entities can preserve their current level of trust in their relationships or exceed their minimum trust threshold to carry out interaction in service-oriented environments successfully.
In proposing the framework, the fundamental assumptions we make for the methodology are stated as follows:

1. We assume that both interacting parties (consumer and provider parties) have been previously involved in a certain number of transactional relationships involving similar criteria.

2. To maintain a positive trust relationship, we assume that both parties are sincere in their efforts to maintain a trusting relationship. The proposed methodology introduces a systematic framework by which two interacting parties can develop and maintain a trust relationship.

3. In the stage of trust relationship development, we propose the involvement of a third-party agent, an intermediate agent or a broker who acts as an impartial entity between the two interacting parties (the service consumer and the service provider). The third-party agent acts as an impartial middleman between the two interacting parties who actively participate in the agreement formulation stage.

4. In the stage of trust maintenance, we also introduce a third-party agent who acts as an unbiased intermediate agent who actively plays an impartial role and carries out the operations. The role of the third-party agent is varied in terms of the methods they use. An example is a third-party agent performing as a service monitoring agent in the service monitoring module. Similarly, the third-party agent plays a role as a service evaluation and decision-making agent in other modules. Often the third-party agent operates on a real-time basis, using both behaviours (mutually agreed and actual) to determine performance discrepancies and provide real-time feedback to the non-complying party

This thesis proposes an SLA-based framework in SDN to provide personalised service delivery with guaranteed QoS services to the consumer to preserve trust relationships. We propose and develop an SLA lifecycle where the SLA starts with the service negotiation and SLA formation stage, followed by the SLA management stage, the SLA violation detection and prediction stage, and the service continuity decision-making stage. The conceptual framework of the developed SLA lifecycle for QoS guaranteed personalised and reliable service delivery in SOA-based SDN is depicted in Figure 4.1.

4.3.1 Service-Level Agreement (SLA) Lifecycle Model

The high-level design of the SLA lifecycle for QoS-guaranteed service-oriented SDN consists of four components. Each lifecycle component can resolve the research issues we identified in the previous section. An overview of these components is given below.

Figure 4.1: *A overview of the SLA lifecycle* by S. Khan, 2022.

1. **Negotiation & SLA Formation:**

The first element of this lifecycle is the service negotiation and SLA formation module, where the most crucial decision is made. The module has four stages: service specification, service provider selection, suitable consumer selection, and service level agreement formulation. The outcomes of this module deliver four significant decisions required to operate each module of the proposed SLA lifecycle. This module will provide service consumers with an opportunity to personalise their services. Moreover, the consumer will be able to select a reputation-based service provider in this module. The detailed design of this module is provided in section 4.3.2.

2. **Service Management**

The service management module is the second element of the QoS-guaranteed SLA lifecycle. A third-party monitoring agent plays a primary role in monitoring the quality committed to service delivery. Therefore, a runtime network monitoring approach is used to gather updated network status information. The outcome of this module is an online monitoring log, and a passive monitoring approach is introduced in this module. Details of the module are discussed in section 4.3.2.

3. **Violation Detection**

Violation detection is the third module of the QoS-guaranteed SLA lifecycle model. Based on the functionalities and objectives of the module, the module is renamed service evaluation and violation prediction. This module consists of two sub-modules: measurement services and condition evaluation services. The network conditions are placed in the framework based on SLA contact in this module. The output of this module is a condition evaluation repository. Details of the module are discussed in section 4.3.3.

4. **Service Continuity**

Service continuity is the last module of the QoS-guaranteed SLA lifecycle model, where all the decision-making tasks are conducted based on the evaluation result. The output of this module may lead to the first module of the lifecycle (service re-negotiation or service-level agreement re-formation) or may continue the services with the present negotiation arrangement. Details of the module are discussed in section 4.3.4.

The detail of each module of the SLA lifecycle model shown in Figure 4.1 is discussed in the following sections.

4.3.2 Solution Overview of the Service Negotiation Framework for Personalised Service Delivery and Enhancing Trust in SOA-based SDN

The framework of service formalisation and negotiation of service requirements assists in achieving personalised service delivery and enhances trust in SOA-based SDN. In the framework, we argue that the negotiation framework could be considered the basis of a trust-building relationship between the service consumer and the service provider. Therefore, for such SOA-based mechanisms to be applied in SDN to achieve personalised service delivery, the following requirements in the first stage of the SLA lifecycle, the service negotiation framework, must be met by both parties.

- ***Requirement 1****: Ensure that the intermediate agent or the broker has a precise and comprehensive understanding of the services requested by the consumer.*
- ***Requirement 2:*** *Assist service providers in making an informed decision on whether to accept a service request.*
- ***Requirement 3****: Assist service users in making an optimal decision in selecting a suitable provider from a list of interested providers.*

This section presents an overview of the service negotiation framework that assists in formalising and negotiating service requirements. Details of the framework are given in Chapter 5. This framework comprises the following five stages:

4.3.2.1 Stage 1: Formulation of the Service Request

In this stage, the intermediate agent (registry/broker) interacts with the service consumer to define, articulate, and prioritise the service requirements. There are four steps in this stage as follows:

Step 1: Determine the Service Requirements:

In this step, the service consumer, in consultation with the intermediate agent, defines and formulate the requirements expected of the service.

Step 2: Quantify the Service Requirements:

Quantifying service requirements is the process of structuring the service requirements in terms of service criteria or service needs and its associated quality description(Fachrunnisa, 2011)

Step 3: Prioritise the Service Requirements:

Service requirements prioritisation determines the level of importance of each service requirement, where less important = 0, moderately important =– 1, important =2, and highly important =– 3.

Step 4: Compose the Service Requirements:

Service requirements composition is arranging the service requirements or service criteria for the requested services associated with a quality description that agrees with both parties. The service requirements composition follows a predefined structure and generates an established document for both parties.

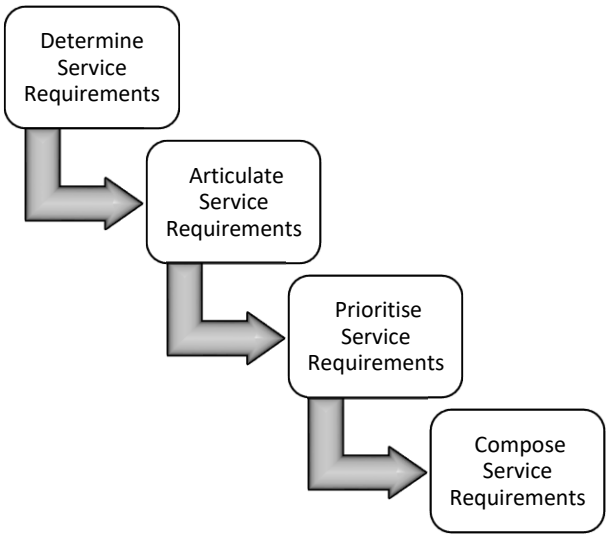


Figure 4.2: *Formulation of Service Request (Pre-Negotiation)* by S. Khan, 2022.

Therefore, the outcome of this joint action represents a clear statement about the service requirements formulation of both parties. The statement contains various elements such as service criteria, service quality descriptors, and level of importance. After collecting the service requirements statement from both parties, the statement is used to negotiate in step 2. By following the top three activities (figure 4.2) (determining service requirements, articulating service requirements, and prioritising service requirements), both parties develop an initial proposal that contains service requirements, quality descriptors, and the importance level of each service requirement. The number of service criteria can be very different for each party ("n" from a service provider and "m" from a service consumer). Therefore, their initial proposal assists in identifying and classifying any conflicting service requirements that both parties can work on further and may require re-negotiated in the next step, the negotiation stage.

4.3.2.2 Stage 2: Initial Proposal from the Service Provider/s

In this stage, the intermediate agent (registry/broker) collates a response from the available service provider/s, addresses the conflicting information in response to the service consumer's requirements, and formulates a response from the service provider to the service consumer. The aim of this stage is achieved in three steps as follows:

Step 1: Translate the service request to a service description template (SDT):

In this step, both parties review and discuss their service requirements, and the intermediate agent translates the formulated service request into a service description template (SDT) (Fachrunnisa, 2016).

Step 2: Identify and address any conflicting or unfeasible requirements:

After the SDT is formed and if, from the perspective of the intermediate agent, the proposed service requirements are either *unfeasible* or difficult to achieve, an adjustment to the service requirements will need to take place to accommodate the interests of both parties. This is done through a process of negotiation to address any conflicting requirement/s.

Step 3: Solicit proposals from the prospective service providers and address any ambiguous responses:

> In this step, the intermediate agent plays the role of a broker and solicits proposals from potential SDN service providers who can satisfy the consumers' needs as defined in the SLA. As a result, the intermediate agent receives offers from different service providers who have the capability and are willing to provide the infrastructure to satisfy the requirements.

Once both parties (Service provider and consumer) have formulated the service requirements, they may engage in an iterative negotiation process to formulate those conflicting requirements. As there is always a possibility that both parties may agree or disagree with the service criteria and their corresponding quality descriptors. Therefore, repeated negotiation may occur until both parties reach a common ground of agreement. The outcome of this negotiation is a service commitment that must be achieved.

The same process repeats until they reach a mutual agreement on behaviour that contains finite services in time slots, the number of checkpoints per time slot, the approach to evaluating each other's performance, and reporting process. The outcome of this negotiation is documented in a contract. Thus, the result of this step is a contract that will provide guidance behaviour to each party to achieve a common goal. The details of the service requirements formulation and negotiation mechanisms are described in Chapter 6.

4.3.2.3 Stage 3: The Service Provider Evaluates the Suitability of Accepting a Request from the Service Consumer

In this stage, the intermediate agent assists the service provider in determining if it should accept the service consumer's request or not by considering its reputation. The intermediate agent passes only those provider agents' who commit to the consumer's request to the next stage. Hence, the objective of this interaction is a) to provide uninterrupted services with guaranteed service quality; b) to deliver differentiated and personalised services to the service

consumer; c) to develop and maintain a level of trust between service consumers and providers. Selecting the service provider is the fourth step of our proposed methodology.

At this stage, we developed a framework to evaluate the suitability of accepting the service request and assist the service provider's decision-making. The following factors are considered for intelligent decision-making and assessing the provider's suitability regarding delivering the requested services.

Factor 1: *Reliability of the Consumer:*

The reliability of the service consumer represents the nature of the past relationships between itself and the contracted service providers. When calculating the reliability of a service consumer, its previous 'n' years of reputational data in (a) the consumer's overall reputation rating *(R)* and (b) the service provider's satisfaction rating of this consumer (Other Provider's Reputation Rating (*OPR*)) are considered.

The generic concept of reputation drives the specific concepts of *service reputation* and *product reputation.* The term *service reputation* is reflected (Hussain, 2006) as aggregating all third-party recommendations from the agents' recommendations. The recommendation is considered as the reply or feedback to a service requestor's reputation query about the QoS that a given service provider is delivering for a given service in a given time slot.

Factor 2: *Duration of the services:*

The duration for which the resources are requested is the second factor considered by the service provider when ascertaining the suitability of accepting a request or not. It depends on the service provider on how it views this criterion.

Factor 3: *Service risk propensity:*

Risk exposure is an important factor to be considered when making an intelligent decision about a service provider's suitability. The cost to be incurred will depend on whether it is

possible to use the available resources or whether additional resources needs to be acquired, which can be evaluated on the risk scale.

This framework's objective is to assist the service provider in evaluating the suitability of accepting a service request. In order to achieve the objective, the three factors mentioned above are considered to ascertain the provider's suitability to accept the service request. We use the "If-Then" rules to receive the output. According to the inputs, 60 variation rules were constructed. The details of the suitability of the provider are described in Chapter 5.

4.3.2.4 Stage 4: Reputation-Based Provider Selection

We propose a reputation-based provider selection framework to assist the service requester in selecting the most suitable service provider. Thus, the consumer has the opportunity to select a suitable service provider among the 'n' number of interested providers on whom they can rely and with whom they feel confident in entering a contract. In other words, the requester company needs to find a reliable service provider to deliver their requested services. In this stage, the intermediate agent (registry/broker) assists the service consumer in selecting the most suitable provider from the available ones.

-

In this stage, the intermediate agent assists the service requestor in selecting the most suitable service provider among the ones who responded with an offer. It does this by determining the suitability of each service provider, which shows how closely associated each provider is with the requestor's needs. The service providers who responded can be divided into established and emerging categories.

Suitability value of an established provider:

Established providers are considered to be those who have provided their services for over two years. In other words, these providers are those who have a reputation value given to them by previous service requestors.

We used the Mamdani method to determine a suitable service provider for the requested services and assist in intelligent decision-making in relation to provider selection from a range of existing service providers. Details of the approach are described in Chapter 5.

Suitability value of an emerging provider:

Emerging providers are considered to be those who are either new or have provided services for less than two years. Compared to their established counterparts, these providers may not have a high reputation value and thus may be disadvantaged when the service requestor decides between this agent and the established ones. To address this, the process by which the intermediate agent ascertains the suitability value for each provider varies, as explained in the following sub-sections.

We utilise the k-nearest neighbour (KNN) technique to classify these agents based on their similarity to existing ones and use their reputation value to infer the emerging provider's value. The KNN algorithm is a classification algorithm that identifies the majority between the k-most similar instances to a given 'unseen' observation. Euclidean distance is used to measure the k-nearest neighbour's profile to enter a predictive decision. We select the top-k nearest neighbours from a set of providers with the maximum number of similarities to a new provider, which is used to make a predictive decision to select a relatively new provider with no previous transaction history. Identifying the top-k nearest neighbours comprises four sub-steps:

a) Similar neighbours discovery
b) Define the best k values
c) Transaction trend of the similar neighbours
d) The decision to approve or reject a new provider

4.3.2.5 Stage 5: The SLA Between the Two Agents is Formulated and Executed

In this stage, the SLA between the service consumer and the selected service provider is formulated and executed. Once both parties determine and agree on all the service requirements that need to be conducted during the interaction, a third-party agent will assist in translating the service requirement formulation and negotiation result into an SLA. Therefore, this is the stage where the SLA construction is commenced. We know that the SLA is a formal agreement between the two interacting parties (service provider and service consumer) with a set of agreed service requirements. Furthermore, In the SLA construction stage, the third-party agent assists in defining the number of checkpoints in each time slot for both parties. Chapter 6 discusses the framework in detail for formalising and negotiating service requirements.

4.3.3 Solution Overview of Framework for Service Management

The service management framework is the second module of the QoS-guaranteed SLA lifecycle framework. Service management is a crucial segment of an SLA. Managing the services committed to being delivered requires continuous monitoring to ensure the contract conditions in the SLA are met. Therefore, continuous service monitoring plays an essential role in determining whether an SLA has been violated (Quillinan et al., 2010). Monitoring is directly related to SLA evaluation, sometimes automated in SLA enforcement mechanisms at runtime without excessive delay (Quillinan et al., 2010). Monitoring an agreement necessitates sporadically testing whether the agreement's terms and conditions have been met.

Service management is coordinated by a special agent or network management and violation detection agent (neutral position). This module will monitor the runtime network and gather updated network status information. This module's outcome will be an online monitoring log.

Continuous monitoring tasks can be observed in two ways: online monitoring and post facto auditing. We have viewed SDN as a service-oriented principle and developed our frameworks

to deliver the network as a service. The online monitoring approach is chosen to perform the continuous monitoring task.

Online Monitoring:

Online monitoring is commonly used to monitor an SLA. Online monitoring is performed in real-time, so securing the monitoring updates against malicious parties is imperative. For our framework, we choose both the proactive and passive service monitoring approaches to perform our network monitoring to manage the services and preserve the SLA.

4.3.3.1 Passive Network Monitoring

Passive monitoring is a monitoring approach that gathers actual network traffic data and analyses it over a specific period. The monitor then studies the analysis and releases the results. Monitoring generally studies the random status of the network. Passive monitoring analyses the real-time network traffic data from specific points in the network rather than analysing test data. Passive monitoring consists of four steps. Each step is interconnected, which means that the outcome of the previous step is the input for the next step.

Step 1: Capture network traffic

The framework includes capturing the runtime traffic data according to predefined time slots. This is a time-intermission-based continuous process to determine if any early intervention is required by making a reasonable adjustment.

Step 2: Identify network traffic type

Identifying the type of network traffic from a set of captured traffic is the second step of the passive performance monitoring framework. The basic idea is to identify or determine the traffic type, such as VoIP traffic, network control traffic, video conference traffic, +and data transmission traffic from the unknown traffic set captured.

We used the deep packet inspection-based method that uses current flow statistical features to determine the traffic type. Therefore, a machine learning-based algorithm (support vector machine) is used in artificial intelligence.

Step 3: Classify network traffic

The third step of the passive network monitoring framework is classifying the identified traffic. Classifying network traffic is an approach that allows the traffic to be organised into traffic classes or categories according to whether the traffic matches specific criteria.

We categorised the traffic type identified from our dataset according to the applications or services at this stage. We classify the traffic in the following three steps:

Step 1: We identified and listed 16 traffic categories available from the dataset.

Step 2: We identified six sub-categories into which all service traffic categories can fit, namely network and control plane protocol, voice application, video application, data application, default forwarding and business irrelevant.

Step 3: We classified the associated sub-categories according to business demand, namely the business relevant category, default category and business irrelevant category.

Step 4: Prioritise network traffic

Traffic prioritisation is a method to add a traffic value to ensure that essential or time-critical traffic flows without delays and assists in service differentiation. Moreover, prioritising the network traffic helps sustain the QoS in the network (Cisco Systems, 2017). We used the traffic prioritisation concept in this framework and developed our method to deliver differentiated services on the SDN services platform.

Based on the urgency and need to sustain the QoS, we prioritised the sub-categories as follows:

Main Category	Sub-category	Priority Number
Business Relevant	**Network and Control-Plane protocol**	1
	Voice Application	2
	Video (Interactive)	3
	(Streaming)	3
	Data Application	4
Default	**Default Forwarding**	5
Business Irrelevant Category	**Business Irrelevant Category**	6

Table 4.1: *Network Traffic Prioritisation* by S. Khan, 2022.

We used Python programming to implement the priority queue method in our framework. Details of this method are discussed in Chapter 7.

4.3.3.2 Proactive Network Monitoring

A proactive monitoring approach is an advanced form of active monitoring by troubleshooting the network to identify potential problems before it impacts the end-user performance. This

approach performs continuous monitoring to search for any indication of issues or any possible issues that are about to occur.

To achieve above mentioned objective (Objective 3), the following mechanism is proposed to perform proactive continuous performance monitoring:

a) Proactively monitor the network using the runtime monitoring approach.
b) Determine and implement the early checkpoint threshold according to the SLA.
c) Determine and implement the SLA threshold according to the SLA.

Proactively monitor the network using the runtime monitoring approach.

The first step of the framework is to monitor the network in real-time to determine the checkpoints that need to be deployed in the monitoring approach. To achieve the first sub-objective, we simulate the SDN network using simulation software.

Determine and implement the early checkpoint threshold according to the SLA.

Early detection checkpoints determine the performance discrepancy threshold of services and the number of points needed to check the performance and apply the service performance discrepancy threshold in the runtime system to identify any performance discrepancies. The intermediate agent needs to determine the number of checkpoints and discrepancy thresholds during the SLA formulation stage. We used the Python programming language to implement the early checkpoint threshold.

Determine and implement the SLA threshold according to the SLA.

SLA threshold checkpoints determine the performance discrepancy threshold of services and the number of points needed to check the performance and apply the service performance discrepancy threshold in the runtime system to determine any performance discrepancies. The intermediate agent needs to discuss and determine the number of checkpoints and discrepancy thresholds during the SLA formulation stage. We used the Python programming language to implement the early checkpoint threshold.

4.3.4 Solution Overview of Framework for Service Evaluation and Violation Prediction.

The service evaluation and violation prediction framework are the third modules of the QoS-guaranteed SLA lifecycle framework. The critical need for developing the service evaluation and violation prediction framework is to ensure impartial and trustful service delivery in SOA-based SDN. The framework is carried out during interaction with a third party or an intermediate agent on which the interacting parties agreed. Thus, the proposed framework facilitates the maintenance of trust using service performance records.

The fundamental hypothesis of our SLA-based framework is that a trust relationship has already been established during interactions between the parties and a trust relationship needs to be maintained by consistently monitoring, examining, and evaluating the interaction. Hence, both parties need to have a transparent picture of the deliverable services by evaluating the deliverable services and identifying any SLA violation during the interaction. A service violation involves penalties, both financial and reputational. Hence, both parties need a framework that detects service violations and can also predict possible violations. Therefore, our proposed service evaluation and violation predicting framework was developed to preserve the SLA and ensure service reliability. The proposed framework consists of two stages as follows:

4.3.4.1 Stage 1: Formulation of the Quantifiable Services

In this stage, the intermediate or neutral evaluation agent interacts with the service consumer and service provider, collecting all the information regarding the deliverable services and translating the information according to predefined measurable metrics that enable the service evaluation process.

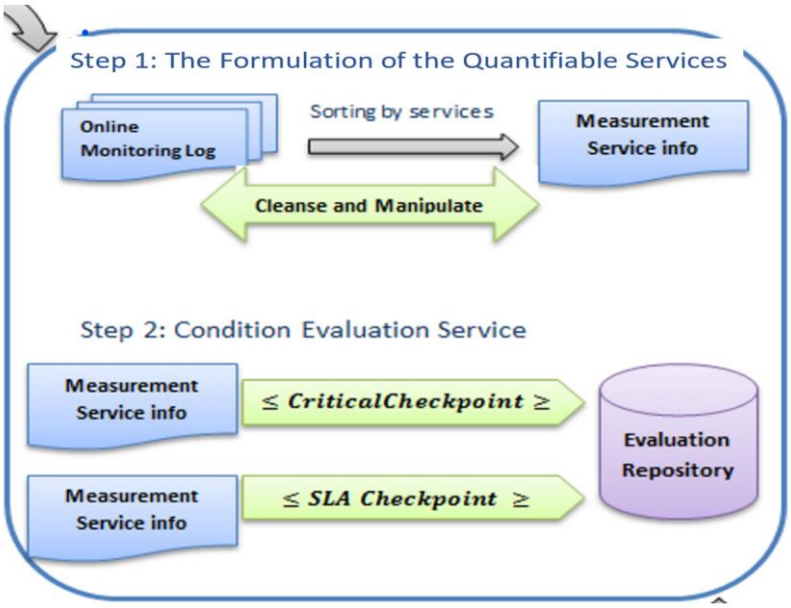


Figure 4.3: *Service evaluation and violation prediction framework* by S. Khan, 2022

The formulation of the quantifiable services approaches assists in turning the raw dataset into quantifiable data according to the evaluation metrics. An intermediate entity or an agent, chosen by both the service receiver and provider, is actively involved in the framework; therefore, the service consumer and provider play no prominent role. As shown in Figure 4.3, the proposed approach consists of the following three steps.

Step 1: Collecting time-series traffic dataset.

Collecting traffic datasets is one of the primary activities of the proposed service evaluation and violation detection framework. We collect the dataset at various times during the interaction and consider the predefined time window.

Step 2: Cleanse the dataset

We cleanse the traffic dataset by identifying data errors and modifying them by changing, updating, or removing data. The process is applied to fix the dataset's incorrect, incomplete, duplicate data.

Step 3: Prepare the dataset according to evaluation metrics

We modify and prepare the dataset accordingly to unitise the dataset for the next stage of the framework. We considered the four performance parameters for the service evaluation and violation detection framework research to evaluate the network service performance.

4.3.4.2 Stage 2: Condition Evaluation Services

In this stage, the intermediate or neutral evaluation agent performs the evaluation process regarding the condition or the QoS parameter metrics committed to in the SLA. The evaluation agent receives the current QoS status and formulates the quantifiable service information from the previous stage. The objective of this stage is to identify and analyse any service discrepancy that may occur as a result of service-level contract violation. The condition evaluation service approach comprises the three following steps:

Step 1 Employ the service degradation critical checkpoint threshold and SLA threshold

The concept of the condition evaluation service is to compare the current network performance with a predefined standard performance and receive a result. To do this, we need to define a standard performance range to be used for comparison purposes; thus, we introduce acceptable performance thresholds in the framework. We used the Python programming language to employ the service degradation critical checkpoint threshold and SLA threshold.

Step 2: Evaluate service performance using the critical checkpoint and SLA threshold

We evaluate the traffic status data, which gives us an accurate picture of the current service performance of the SDN with the defined checkpoint threshold to ascertain how far the service degradation level is from the predefined checkpoint threshold level. To evaluate service performance using the framework, we used the Python programming language to implement the approach.

Step 3: Applying the performance discrepancy condition and marking the traffic

After evaluating the service performance, we employ a condition to assist in ascertaining the overall idea of the current service performance status. This step assists in processing the decision-making of the following framework.

4.3.5 Solution Overview of Framework for Service Continuity Intelligent Decision-Making.

The service continuity decision-making framework is the fourth module of the QoS-guaranteed SLA lifecycle framework. There is a critical need to develop the service continuity decision-making framework to provide an honest and trustful decision-making approach in SOA-based SDN. A third party carries out the framework in the presence of the consumer party during the interaction, especially at the midpoint of the interaction to which the interacting parties agreed. Thus, the proposed framework facilitates both parties' impartial assessment of the entire interaction and assists service continuation decision-making.

We have argued in this thesis that a positive trust relationship facilitates an impartial association of service continuity decisions if service reliability or confidence is achieved. Therefore, we require a framework that can impartially assess the deliverable services which operate with completeness, honesty, and security and assist in decision-making. This framework provides a genuine representation of the outcome of the objective of delivering reliable services and preserving the trust relationship. This section provides an overview of the

framework, and the details are available in Chapter 8. The proposed framework consists of three stage s.

stage *1: Summarisation of reaching the threshold*

The *summarisation of reaching the threshold* stage of the service continuation framework considers the results of the previous framework (service evaluation and violation prediction framework) and is analysed further in stage 2.

stage *2: Employing the maximum count of acceptance rule*

In this stage, the intermediate or neutral evaluation agent performs another stage of the evaluation process in terms of the condition for a maximum count of acceptance rules which were predefined and committed to in the SLA. This stage aims to analyse, identify, and employ the acceptance value against all the service discrepancies that may or may not lead to a service-level contract violation.

stage *3: Evaluation and decision obtained*

In this stage, the intermediate agent evaluates the results obtained from stage 1 (*summarisation of reaching the threshold*) against the stage 2 rules (*employing the maximum count of acceptance rule*) and makes a decision on whether the service consumer should continue their business transaction with their current provider or should consider engaging a new provider to support and deliver the services according to their business needs.

We validate the framework using the Python programming language, and the implementation details are discussed in Chapter 8.

4.3.6 Solution Overview to Validate the Methodology

In this thesis, we make use of simulation-based experiments to validate the QoS management methodology using a service-oriented lifecycle-based framework in service-based SDN environments. Moreover, these experiments are used to determine the reliability or suitability of an entity in a given context in a defined time slot by employing the reputation data-driven framework. We developed a reputation rating time series dataset and validated the dataset using AI-based methodologies. We engineered the framework to validate the SLA-based QoS framework in the following four sub-frameworks.

4.3.6.1 Sub-Framework 1: Service Negotiation Framework

From the above discussion, we know that the service negotiation framework assists both the service consumer and the service provider in selecting the appropriate party to execute their SLA; therefore, the framework consists of five stages, and associated sub-approaches are used to validate these stages.

1. Reputation rating time series dataset development is a significant contribution to this framework. We have developed a reputation rating time series dataset with 500 service provider reputation ratings and used the Python programming language to develop the dataset. We used root mean square error (RMSE) to predict the errors by employing residual standard deviation. Details of the reputation rating synthetic dataset and validation are discussed in Chapter 5.

2. The service provider evaluates the suitability of accepting a service request: this approach considers three factors (reliability of the consumer, duration of the services, and service risk propensity), and the implementation overview is as follows.
    a) The reliability of the consumer is determined using an equation that we developed in our thesis and is considered Factor 1. The equation is the overall reputation rating data of the consumer and other providers' satisfaction ratings of this provider. Then, we developed a reliability scale where the maximum value is five and the minimum value is 0. After scaling the reliability, we utilised the Mamdani method, a fuzzy logic analysis approach using MATLAB to articulate the input variables.

b) The duration for which the resources are requested is determined by developing a service duration rating scale where 0 to 3 months is considered too short and 24 months or more is considered too long. This is considered Factor 2. Then, we utilised the Mamdani method, the fuzzy logic analysis approach using MATLAB to articulate the input variables and assist in intelligent decision-making.

c) The duration for which the risk exposure of the service provider accepting the service request is calculated using an equation and a possible risk rating scale where the $0.00 to $n1 expense range is considered a low-risk rating and $n3 to $n4 is considered a high-risk rating. This is considered Factor 3. Then, we utilised the fuzzy logic analysis approach based on the Mamdani method in MATLAB to articulate the input variables and assist in intelligent decision-making.

3. The aforementioned three factors determine the provider's suitability to accept the service request. To perform the suitability calculation, we used the fuzzy inference system (FIS) and employed If-Then rules. We developed 60 variations of the If-Then Fuzzy association rules using MATLAB and received the output. We mapped the input variable strength and generated the outputs. Based on our fuzzy association rules, company reliability obtained the highest strength and mapped the input variables accordingly.

4. To assist the service consumer in determining a suitable service provider, we developed two frameworks for emerging and existing providers.
    a. For the existing provider, we used the Mamdani FIS method to accelerate the decision-making process. We used MATLAB for this implementation task.

    b. To assist the service consumer, identify a suitable emerging service provider which we assume does not have a reputation rating, we used the KNN algorithm to validate the framework following the Euclidean distance calculation.

- Firstly, we employed a similar neighbour's discovery by following the nearest neighbour (NN) equation and determining the nearest neighbour of the emerging provider.
- Secondly, we determine the best k value, where k is the parameter of the number of nearest neighbours that gives the best outcome and is included in the research.
- Thirdly, we determine the transaction trend of similar neighbours using our transaction trend equation.
- Fourthly, we determine the transaction trend of the emerging provider employing our developed k neighbour-centred transaction trend equation.

Thus, the consumer evaluates and selects the emerging provider using our decision-making equation. The Python programming language is used to implement the algorithm. We also used the Mamdani method for the decision-making process and implemented it in MATLAB.

4.3.6.2 Sub-Framework 2: The Service Management Framework Employs Proactive and Passive Continuous Service Performance Monitoring

The service management framework consists of two service monitoring frameworks; therefore, we validate the framework using two different approaches. The validation of each approach consists of several steps.

1. Proactive Service Monitoring:

   To develop the proof of concept of the proactive service monitoring framework, we used the GNS3 emulation tool to create the network architecture.

   a. We built virtual machines (VMs) and incorporated these VMs into the GNS3 environment.
   b. We configured the Zabbix appliance in GNS3 to assist in the real-time monitoring of the entire network. Zabbix accesses the front-end view using a browser; therefore, we configured it to access real-time network monitoring using any computer on the network.

c. We set up the early checkpoint threshold alert in the Zabbix monitoring appliance using a trigger alert.
d. We set the SLA threshold checkpoint using a more complex trigger alert in Zabbix.
e. We tested the success of the proactive monitoring approach by making some of the devices fail and analysed how the Zabbix monitoring appliance triggered the alert.

2. Passive Service Monitoring

The passive service monitoring approach depends on the simulated network we configured in the proactive service monitoring stage. We perform the validation of the passive performance monitoring in the following steps.

a. We used Wireshark to capture the traffic data from the simulated running network. We take a sample from the traffic dataset to analyse further.
b. To identify the network traffic type from the sample set, we used a machine learning (ML)-based algorithm, the support vector machine (SVM). A rapid miner is a software tool used to implement the SVM algorithm.
c. We developed a traffic classification model to classify the network traffic, and the Python programming language is used to implement the traffic classification model.
d. To ensure QoS delivery, we developed a network traffic prioritisation model and implemented the model using the Python programming language.

4.3.6.3 Sub-Framework 3: The Service Evaluation and Violation Prediction Framework.

The service evaluation and violation prediction framework consist of two stages. This section overview the validation of the two stages.

1. Formulation of the quantifiable services.

As previously mentioned, we collected a real-time traffic dataset from our configured network simulation. The network simulation is built using VMware and GNS3 with the Zabbix appliance monitoring tool. We collected the time series dataset using Zabbix. We

used the Python programming language and Excel to cleanse and modify the dataset according to our predefined performance metrics and analysis.

2. Condition evaluation services

We used the Python programming language to program the code to evaluate the runtime services according to our developed service evaluation measures. We employed our defined performance measurement metrics and determined the maximum and minimum values based on the services. Our program allows the evaluator to enter their SLA-based performance values. Therefore, our developed system can compare the current service performance values collected from the real-time dataset with the evaluator-defined SLA-based performance dataset and obtain the results.

4.3.6.4 Sub-Framework 4: The Service Continuity Decision-making Framework

The service continuity decision-making framework consists of three stages. An overview of the framework validation is provided below, and details of the validation are available in chapter 8.

1. *Summarisation of reaching the threshold*
   We used the Python programming language to summarise reaching the critical checkpoint threshold and the SLA checkpoint threshold.

2. *Employing the maximum count of acceptance rule*
   In this stage, we used the Python programming language to employ the maximum count of acceptance rule in the code.

3. *Evaluation and decision obtained*
   To evaluate the decision obtained, we used If-Then conditions and the Python programming language.

This thesis efforts to address the fundamental need for QoS management by proposing a methodology with simulation-based experimentation to maintain trust in an SOA-based SDN environment. This thesis has six objectives. The first objective (objective 1 and objective 2) focuses on the formulation of service negotiation and the SLA. It can be denoted as the trust-building stage, and the remaining three objectives (objective 3, objective 4, objective 5 and objective 6) focus on performance evaluation and assist in rational decision-making, which can be denoted as the trust maintenance stage. The SLA lifecycle frameworks to guarantee QoS in SOA-based SDN forms with six states are as follows

a) The definition of the critical terms is developed in this thesis, which needs to address the research question and develop solutions (objective 1).
b) Service performance negotiation of the SLA lifecycle approach intelligently personalises the services and delivers them to the consumer (objective 2).
c) Service management framework of the SLA lifecycle approach intelligently and monitors the SDN services and provides a predictive decision of pro-active service violation (objective 3).
d) The service evaluation and violation prediction framework of the SLA lifecycle approach measures the QoS guarantee level in terms of the reliability of the services (objective 4).
e) The service continuity decision-making framework of the SLA lifecycle approach assists in intelligent decision-making for service continuity (objective 5).
f) An experimental prototype to validate the aforementioned methodological frameworks.

Each objective and the associated framework for the objective are presented sequentially from Chapter 5 to Chapter 8. Each chapter includes the framework details and a discussion of the experiment implementation, the results and the framework.

4.4 Conclusion

This chapter represented a standard formal definition of the trust evolution model concepts and described its three co-related stages: trust building, trust maintenance, and trust decline. Moreover, we presented the characteristics and features of each trust evaluation stage. In addition, we provide a high-level solution overview for each of the six research issues identified in Chapter 3. Finally, we outlined a solution overview to the fundamental problem addressed in this thesis. In the next chapter, we propose the framework for service negotiation for personalised service delivery in SOA-based SDN, which was identified in this chapter as being the first step of our methodology to build the trust relationship between the service consumer and service provider.

4.5 References

Chang, E., Hussain, F., & Dillon, T. (2006). *Trust and reputation for service-oriented environments: technologies for building business intelligence and consumer confidence*. John Wiley & Sons.

Cisco Systems, I. (2017). *Cisco EasyQoS Solution Design Guide* (1.6 ed.). Cisco.

Fachrunnisa, O. (2011). *A methodology for maintaining trust in virtual environments*, Curtin University].

Fachrunnisa, O. (2016). A Framework for Service Formalization and Negotiation for Trust Maintenance in Digital Environments.

Hussain, F. K. (2006). *A Methodology for Trust Management in Service Orientated*

Environment, Curtin University of Technology].

Quillinan, T. B., Clark, K. P., Warnier, M., Brazier, F. M., & Rana, O. (2010). Negotiation and monitoring of service level agreements. In *Grids and Service-Oriented Architectures for Service Level Agreements* (pp. 167-176). Springer.

Chapter 5

SERVICE NEGOTIATION FRAMEWORK

5.1 Introduction

In network management, software-defined networks (SDN) assist in the efficient management of the network infrastructure more so than traditional mechanisms. It uses standardized application programming interfaces and centralized mechanisms to divide, manage and control the whole network through the control plane(Jain & Paul, 2013). These features also enable network managers to customize and provide parts of the network infrastructure to different service users as required. This paper focuses on SOA- based SDN. SOA is a software design architecture where the principle is to provide a dynamic composition of service/s that is loosely coupled, abstracted, and reusable (Khan, 2020). These aspects of an SOA-based SDN make the network architecture flexible and adaptable enough to utilize in a more dynamic setting in response to changing environmental conditions. However, before such advantages are achieved, there are several challenges that need to be addressed. One of these is the ability of the service consumer/s network infrastructure provider/s to negotiate and agree on the quality of service (QoS) specifics required of the service. In other words, this ability requires the service consumers and network infrastructure providers to personalize the particulars of the service required and to be provided, respectively, from their perspectives where the "one size fits all" idea is not applicable. By offering personalisation, both the requestors and providers make the services being received and delivered as needed leading to enhance trust between them. This will be a critical factor when future service agreements involving these agents need to be formed.

Researchers in the literature have proposed frameworks to personalize service delivery in SOA to enhance trust among the interacting agents. We also observed that the service negotiation framework enables an opportunity for personalizing the consumer's requirements which we compare with the trust-building stage (Fachrunnisa, 2011; Hussain, 2006). Most of the work consists of an independent agent, as a broker, who liaises with the service consumer and service provider to utilize a reputation-based selection process that facilitates intelligent decision-making. We argued that the negotiation framework could be considered the ground of a trust-building relationship between the service consumer and the service provider. Therefore, for

such SOA-based mechanisms to be applied in SDN to achieve personalised service delivery, the following specific requirements need to be met:

Requirement 1: Ensure that the intermediate agent or the broker has a precise and comprehensive understanding of the requested services by the consumer (hereafter considered R1).

Personalisation, which occurs during the service negotiation process, facilitates a consumer's service request more efficiently and effectively. However, for this to materialize, an essential activity for the broker is to determine the service requirements from the requestor, identify and address any *conflicting or unfeasible* requirements and prioritise them according to the requestor's needs. If these factors are not achieved, then it is quite possible that there may be a mismatch between the recommended network provider in terms of the QoS that it can provide and what the service consumer needs according to its requirements.

Requirement 2: Assisting the service providers to make an informed decision as to whether to accept a service request or not (hereafter considered R2).

In an SOA-based architecture, there may be different service consumers that need the available resources of a specific provider for the required time period. Furthermore, the provider committed to using its available resources for different consumers at a period in time in the future. Thus, a certain service provider needs to consider these factors when evaluating the suitability of accepting a request from a specific service consumer. If these are not considered, then it is possible that the infrastructure to which the provider committed to a requestor may not be available when needed, thereby resulting in a violation of the agreed SLAs.

Requirement 3: Assisting the service users to make an optimal decision in selecting a suitable provider from a list of interested providers (hereafter considered R3).

As may be the case with service providers, a service requestor too may have to choose from multiple service providers to fulfil its requirements. However, to ensure the personalised delivery of services, the service user needs to be sure that the provider it selects is suitable based on its business needs.

In the domain of SOA-based SDN, existing provider selection approaches do not consider these requirements and hence may not result in making an informed selection, both from the perspective of the service consumer and service provider. This paper addresses this gap by proposing a framework that can provide personalised service delivery in SDN. For the service provider, the proposed approach uses a reputation-driven intelligent framework to decide whether to accept a request from a specific consumer or not. For the service consumer, the proposed approach assists in clarifying the service requirements and using a reputation data-driven intelligent framework to select the most appropriate service provider with which to interact. In doing so, it will assist in enhancing the trust relationship between the agents, which will be utilized in forming future service agreements.

5.2 Related Research

Several service negotiation frameworks have been developed in the literature to provide personalised service delivery in different domains. Specific to the context and aim of this paper, we classify the approaches that we discuss into three dimensions to determine if they assist in achieving personalised service delivery in SDN to enhance trust. The three dimensions are *service negotiation frameworks in SDN, personalised service delivery in SDN,* and *service negotiation to build trust*. Relevant approaches in these dimensions are discussed in the following sub-sections.

5.2.1 Service Negotiation Framework in SDN

We reviewed the existing research to determine the insight based on the criteria detailed in Table 5.1. (Chieng et al., 2005) propose an SLA-driven service provisioning architecture to

enable flexible and quantitative SLA negotiations for providing network service. A well-developed agent-mediated network service prototype is demonstrated with a dynamic SLA negotiation scheme between user and network provider agents. It introduces the following four SLA negotiation schemes, namely (a) Bandwidth Negotiation at Resource Limit (BNRL), (b) Guaranteed Session Start Time Negotiation with Delay (STN-D), (c) Guaranteed Session Duration Negotiation for both Session Cut Short (SDN-CS), and (d) Temporary Session Bandwidth Drop off (SDNTBD). Negotiation occurs according to one of the above schemes when the network's performance disintegrates. The impact of the negotiation schemes is measured in three dimensions: service availability, network utilization or revenue generated, and user satisfaction to reduce rejection probability. The approach can predefine and allocate network resources to achieve the targeted performance according to the user satisfaction criteria. However, the research has a limited ability to assist in any decision-making regarding the requirements (R2 and R3) that we argued in the previous section.

(Gomes et al., 2014) propose an approach that identifies the best-suited protocol target among those offered to fulfil the demand according to the client's requirements. It determines which protocol is the most suitable for SDN and virtual networks (VMs) and should be used for negotiation. The design of this approach proposes a similarity model and a similarity metric that utilizes two significant aspects, namely Virtual Network Negotiation, where the model can categorize the protocols to customize to the virtual network SDN; and the metric proposed to evaluate the similarity between the protocols to determine the most suitable. The proposed similarity model enhances the providers' competitiveness and allows the client to engage in informed decision-making to determine the most suitable provider who can fulfil their requirements by considering a list of the protocols offered by the providers. The proposed approach provides an opportunity to select a suitable provider; however, there are no opportunities for a provider to select a suitable service request. Moreover, the overall approach is viewed as a service-oriented approach; however, minimal actions illustrate an intermediate agent in the paper.

(Gomes et al., 2017) propose a mechanism to manage a virtual software-defined network (VSDN) to negotiate, deploy and adapt the network resources according to the current state of the network infrastructure and the SLA delineation. The research argues it is acceptable to

define SLA measurable or unmeasurable parameters and determine the most suitable parameters for VSDN, provide guarantees that the ISP attends to these parameters and improve the usage of network resources and the energy efficiency of the ISP. The research claim is to ensure the quality of the service of VSDN; however, this research does not meet any of the requirements (R1, R2 and R3) that were argued in the previous section.

On the other hand, a dynamic and flexible framework is proposed to select the most suitable cloud service provider using the ranked voting method (Baranwal & Vidyarthi, 2014). This ranked voting method uses application-dependent and user-dependent metrics to rank the cloud providers. In this approach, the metrics are considered as a voter, while the cloud providers are candidates. The research, which is the proposed metrics-based approach, is developed in the cloud domain, which delivers the platform as a service. The research argues for an effective approach to service provider selection while the service request acceptance decision-making (R2) remains unattended.

(Gomes et al., 2013) developed an SLA re-negotiation-based approach to adjust the allocation of network resources to be as close as possible according to the virtual network's demand. This assists the proposed approach to prevent resource idleness and the loss of QoS experience. However, this research only considered the SLA re-negotiation approach rather than the service pre-negotiation process before formulating an SLA. Therefore, the service provider, consumer, and intermediate agent have minimal scope to determine the service requirements during the interaction. Moreover, the approach does not assist any decision-making in terms of a service provider or service consumer request selection.

(Körner et al., 2014) proposed an implementation-oriented conjoint technique (WS-Agreement standard and Open Flow standard) for VMs in the cloud. The research primarily focuses on automated negotiation and creating SLAs for network services to ensure QoS in terms of bandwidth between VMs in the cloud where the SDN architecture is located as a physical network in the backend. We define the QoS requirements of the network and negotiate the service-level objectives to achieve the following capabilities: defining QoS requirements of the network, negotiating service-level objectives based on current network utilization, creating an SLA for a cloud-based network, and establishing a QoS overlay in an open-flow network on

SLA. The developed SLA framework, named WSAG4J, enhances the capabilities of the above approach by allowing the implementation of the WS agreement and the WS negotiation as protocol. Subsequently, the experiment results show that this framework can assist through simpler network management and network utilization calculation and predict the available capacity concerning all end-to-end overlays. The framework WSAG4J participates in three unified actions: get template action, negotiate action, and create agreement action. To perform the above actions, there is minimal or almost no activity of the intermediate agent demonstrated in the paper. Moreover, the research primarily concentrates on a single QoS parameter (bandwidth) instead of a list of critical parameters. In addition, the approach does not contribute to any decision-making in terms of a service provider or service consumer request selection.

5.2.2 Personalised Service Delivery in SDN

(Gharakheili et al., 2014) propose an SDN-based self-customizable architecture comprising a cloud-based front-end portal and SDN-based backend APIs. The architecture is beneficial for subscribers to improve streaming video (YouTube) quality and video conferencing such as Skype. The paper argues that SDN provides a way for households to automate self-customization, while cloud-based delivery simplifies subscriber management. The concept of personalizing services is used in home network experiences using SDN, with a few different objectives in delivering personalizing network services that we argued in the previous section. The framework has no discussions about the activities of the intermediate agent; however, the framework aims to deliver the services. Moreover, the research does not assist in any decision-making as indicated in the previous section referencing requirements R2 and R3.

A provider-based personalised SLA framework is proposed to prevent SLA violations (Walayet Hussain, 2016). This paper develops an optimal SLA formation architecture from the provider's perspective. This research assists the provider in evaluating the consumer's reliability in committing to the SLA, optimizing the use of available resources, and obtaining a maximum return. Moreover, this research helps the provider monitor the consumer's post-interaction and uses this information in the pre-interaction stage to minimize the risk of violating the SLA. Regarding our predefined requirements discussed in the previous section, there are currently

very few activities defined for the third-party or intermediate agent, and no framework is proposed by which to select a suitable provider in terms of the defined service requirements.

5.2.3 Service Negotiation to Build Trust

To ensure trust management between the service provider and service consumer, selecting a suitable service provider is one of the critical tasks. By understanding this, Devi et al.(2016) propose a service provider selection approach from a list of providers in a cloud environment using quantitative and qualitative factors and feedback. The research compares existing research in the cloud, which follows the service-oriented principles in managing trust. The research primarily identifies studies using qualitative and quantitative metrics to select a suitable provider. However, to manage a trust relationship, the capabilities of choosing a suitable contract by an intermediate agent and the service provider during the interaction are overlooked.

(Li et al., 2015) propose a trust-aware service brokering-based approach (T-broker) for multiple cloud collaborative services. The authors construct a brokering system that acts as middleware for efficiently matching providers to deliver different users' requests that assist in trust management. The five critical components of this architecture are (a) sensor-based service monitoring, (b) virtual infrastructure manager, (c) SLA manager and trusted resources matching to find the best trustworthiness resources, (d) hybrid and adaptive trust computation model, and (e) service feedback and aggregation. Their experiment demonstrates that T-broker generates excellent outcomes in many typical cases and robustly deals with various dynamic service behaviours from multiple cloud sites. In terms of these three requirements in the SOA-based mechanism to achieve personalised service delivery, the T-broker framework employs a trust-aware brokering architecture responsible for managing trust and scheduling resources. However, the approach does not assist decision-making activities regarding these three requirements.

(Garg et al., 2011) propose a service management index cloud (SMICloud) approach that allows the users to compare and select the cloud services offered by the providers in terms of their priority sets and several cloud service dimensions according to demand. This approach

addresses two challenges, namely (a) measuring various attributes using historical and promised values to find the actual value by proposing a framework named the service measurement index (SMI) and (b) successfully ranking the service provider attributes using their proposed framework named service ranking using the analytic hierarchy process (AHP). The SMICloud framework can compare the different cloud providers and is able to make optimal decisions regarding cloud provider selection. Both quantitative and qualitative parameters have been identified and used for this model. The AHP ranking mechanism assists multiple criteria decision-making. A SMICloud broker is actively involved in collecting customers' requirements and discovering and ranking suitable services using other components such as the SMICalculator and ranking systems. Therefore, the framework does not have an opportunity to determine a suitable contract or consumer request before formulating the SLA. Additionally, the framework is developed targeting cloud services only; no experiment has been conducted on SOA-based SDN services.

(Ghosh et al., 2014) propose an approach combining trustworthiness and the competence-based approach to select a cloud service provider framework (SelCSP). The proposed approach assesses the interaction risk for service requesters or consumers. Moreover, the research focuses on quantitative and quantitative factors to estimate the providers. The quantitative factors include trust, competence, and reputation. This framework calculates the trustworthiness using the interaction of the service provider and the provider's rating values, which is the consumer's feedback. The following three fundamental parameters are used for computations: a) trust estimation, b) competence estimation, and c) risk calculation. The paper contributes to a broad area to build and manage trust relationships for the service-oriented cloud that can significantly and effectively deal with service providers. However, no proposed method will help to choose a suitable consumer request.

Study	**Algorithm/ Architecture**	**Contribution**	**Area of Applications**	**SOA Principles**	**Implementation**	**Requirements**		
Service Negotiation Framework in SDN						**R1**	**R2**	**R3**
(Chieng et al., 2005)	SLA Architecture	Dynamic negotiation scheme	Internet Services	√	Fujitsu Phoenix Open Agent Mediator (O.A.M.)	×	×	×

(Gomes et al., 2014)	Negotiation protocol	Protocols similarity model for SDN and virtual networks negotiations	SDN.	√	Theoretical approach	×	×	√
(Gomes et al., 2013)	SLA re-negotiation framework	Optimize utilization of resources and provide QoS	Internet	√	Not specific	×	×	×
(Gaurav Baranwal and Deo Prakash Vidyarthi, 2014)	Provider selection using ranked voting	A ranked voting method to select the best cloud service, provider	Cloud	√	Theoretical	√	×	√
(Körner et al., 2014)	Automated negotiation and SLA creation framework	WS-Agreement (Negotiation) Open Flow conjoint approach for guaranteed QoS between VMs in the cloud	SDN	√	Mininet Tool	×	×	×
(Gomes et al., 2017)	Performance negotiation in VSDN	A mechanism to negotiate, deploy and adapt the VSDN performance requirements	VSDN	√	Not defined	×	×	×
Personalize Service Delivery in SDN.								
(Gharakheili et al., 2014)	Personalize improvement of streaming video	Cloud-based front-end portal and SDN-based backend API	Cloud-based SDN	√	Prototype-based implementation	×	×	×
(Walayet Hussain, 2016)	OPV-SLA	Provider-based optimized personalised viable SLA framework to prevent SLA violation	Cloud services	√	Case study-based implementation	×	√	×
Service Negotiation to manage the trust								
(Li et al., 2015)	T-Broker	Service brokering architecture T-broker using hybrid and adaptive trust model, lightweight feedback mechanism	Cloud Services	√	Prototype	×	×	×
(Garg et al., 2011)	SMICloud	Service Management Index Cloud (SMICloud) approach	Cloud Services	√	Case study-based demonstration	√	×	√
(Ghosh et al., 2014)	SelCSP	Facilitate selection of cloud service provider	Cloud service	√	Case study-based demonstration	×	×	√
(Devi et al., 2016)	Provider selection	Comparative analysis to find qualitative and quantitative approaches	Cloud service	√	Review based research	×	×	√

Table 5.1: Comparative analysis of the existing literature

The above literature survey shows little research on SLA-driven service provisioning in SDN. As a result, the existing research does not define how service negotiation strategies can occur amongst providers and service consumers by a third party to achieve SOA principles. Existing research proposes a framework to assist in the post-negotiation stage from a provider's perspective. However, minimal research focuses on the pre-negotiation stage before finalizing the SLA. From a consumer's perspective, minimal research assists them in making informed service provider selection decisions. In contrast, some researchers manage and evaluate trust; however, they do not investigate provider selection from the perspective of SOA-based SDNs. Therefore, personalised service delivery in SDN is an open research question.

5.3 ABC Education Pty. Case Study

This section details a case study and discusses the issues associated with our proposed service negotiation to provide a personalised service delivery framework. Let us consider the following example:

ABC Education Pty. is an education provider company looking for effective network services that can deliver and continue the education process without interruption. As a few reputed SDN service providers are in the current market, it is difficult for ABC Education to find a trustworthy and suitable provider who can fulfil their needs. From this point of view, ABC Education decides to involve the intermediate agent in communicating with a service provider and finding the best fitting provider and services for them to receive their required services. ABC Education enters into contact with the SMART Mediation intermediate agent.

Let us consider that SMART Mediation is an intermediate agent that assists in collaborating and generating the interaction between the service requester and providers. We also assume that SMART Mediation has its repository and that the current SDN service providers are registered with SMART Mediation, which enables them to share their company profile with SMART Mediation. On the other hand, if a service provider is relatively new to the current market, it can also register with the intermediate agent as a new provider. Therefore, the new

provider has a very limited business profile to share. Table 5.2 shows the details of each party's position in the scenario.

Service requester/Consumer	**Intermediate agent**	**Service provider**
ABC Education Pty.	SMART Mediation	XQuery Ltd.

Table 5.2: Interacting parties in the case

Let us further discuss the service requester or consumer. In this scenario, ABC Education, the service requester, consecutively runs several institutes in various states in Australia. To run several courses simultaneously, it is essential that there is a demand for their network. Furthermore, the students complete online professional assessments where the availability of an uninterruptable network is critical. Furthermore, due to the nature of this work, network availability is the key, and secure, reliable, and high-quality networks and services are crucial. To meet their service needs, the following considerations need to be taken into account: (a) ABC Education is looking for a reliable provider; (b) ABC Education also wants to avoid any future dissatisfaction that leads to an unhappy relationship; (c) ABC Education wants to prevent any future violations of the SLA commitment to avoid paying the penalty and losing financial benefits.

On the other hand, XQuery Ltd is an SDN service provider registered with intermediate agent SMART Mediation. Let us assume that XQuery Ltd is a medium-sized organization and would like to deliver its services to a requesting consumer, such as ABC Education. Based on their transaction history, it can be seen that XQuery Ltd had never worked with ABC Education before. As a result, there are no financial or service-oriented transactions.

The provider company has the infrastructure, hardware, software, and expertise to deliver its services. The consumers request their ‘required resources, which are those resources that are essential for the consumer to run their business, and 'marginal resources, which are those added resources that are stored as a reserve and used in case of an increase in business demand. We assume that a large service provider has unlimited resources and is capable of supporting

various services. In contrast, a small service provider has limited resources and is required to use them optimally. As a small company, the difficulties they may face are as follows: (a) it is a significant challenge for a small service provider company to make the best use of their resources and benefit from doing so; (b) it is a challenge for small provider companies to utilize their essential and marginal resources to deliver a requested service; (c) it is a crucial decision for the provider company to use their maximum resources, taking a maximum risk to deliver the services to a consumer who may be very much unknown to the provider; (d) it is challenging for a provider to agree to the maximum resources decision without knowing the consumer's profile with their reputation score.

ABC Education requests SMART Mediation's involvement, and ABC Education must share its company reputation profile. ABC Education may be an old or new consumer for the agent. Let us consider that ABC Education has provided education all over Australia for the last ten years; however, they have never worked with SMART Mediation before. Hence, ABC Education is a new consumer for the agent, and as so, SMART Mediation has little understanding of the company. When ABC Education registers with them and requests assistance, the agent accesses their company profile and collects essential information they need to share with the potential service provider while forming an agreement.

To develop a healthy relationship among the trusted parties and trusting parties, (a) each party must choose an appropriate party that will meet their requirements and develop and maintain a long-term relationship to deliver the best possible outcome; (b) to avoid any dissatisfaction among the two parties and to develop a good relationship with every party, ABC Education is looking for an intelligent decision-making approach to select the most suitable provider by analyzing the company's circumstances.

From the above discussion, we formalize each party's requirements to develop and maintain a trust relationship as follows:

1. Since each party wants to develop and maintain a trusting relationship, they need to determine their business requirements and articulate them clearly. Once the business

requirements have been articulated, subsequent negotiations between them can take place.

2. A guideline or transparent standard is required to evaluate and benchmark the services.

3. A service criteria statement/ requirement formulation is required to understand each party's interaction requirements, including services. The service requisite formulation framework will permit ABC Education and Xquery Ltd. to formalize their requirements and standards from the relationship.

4. A service negotiation framework is in demand to build an agreement on service requirements. The negotiation framework assists in minimizing any differences in the requirements between all interacting parties and will have provision to discuss any conflicting requirements that lead to finalising an agreement.

5. Selecting a suitable and reliable provider from a service provider list is essential to developing and maintaining trust. Finding a suitable provider for the consumer to build their confidence in delivering their required services is challenging. The reputation rating of the provider may strengthen the selection; however other factors are not unavoidable. A provider selection framework is also on the demand list for the consumer to make an intelligent decision.

6. Every provider has its scope or limitations to deliver the best services to the consumer. Considering their boundaries, it is important for their business survival to make the best decision as to whether the provider should deliver the services that have been requested. This is an intelligent decision that the provider needs to make and confirm before finalizing an agreement. An intelligent decision-making framework is necessary for the provider to make the decision.

5.4 Proposed Service Negotiation and Service-Level Agreement Formulation Framework

This section presents our proposed reputation-driven intelligent framework to provide personalised service delivery in SDN. The proposed framework assists both the service consumer and the service provider in selecting the right party to execute their SLA. This is after ascertaining the criteria required from the service, articulating, and prioritising them before forming the SLA to be committed to. For the service consumer, the proposed framework assists in intelligently selecting a suitable service provider from the different available ones, and for the service provider, it assists in intelligently deciding whether the service request should be accepted or not considering the consumer's reputation and the provider's circumstances, such as their available resources. Considering both the service consumer's and the service provider's perspectives enables both parties to assess each other on different aspects before a decision is made to build a trusting relationship between them. This also means that the service consumer has a broader scope to receive personalised service delivery from the most suitable service provider. As shown in Figure 5.1, the proposed framework consists of five stages as follows:

Stage 1: Formulation of the service request:

In this stage, the intermediate agent (registry/broker) interacts with the service consumer to define, articulate, and prioritise the service requirements.

Stage 2: Initial proposal from the service provider/s:

In this stage, the intermediate agent (registry/broker) collates a response from the available service provider/s, addresses the conflicting information in response to the service consumer's requirements, and formulates a response from a service provider to the service consumer.

Stage 3: The service provider evaluates the suitability of accepting a request from the service consumer :

In this stage, the intermediate agent assists the service provider in determining if it should accept or not the service consumer's request by considering its reputation and circumstances, such as available resources. The intermediate agent passes to the next stage only those provider agents who accept committing to the consumer's request.

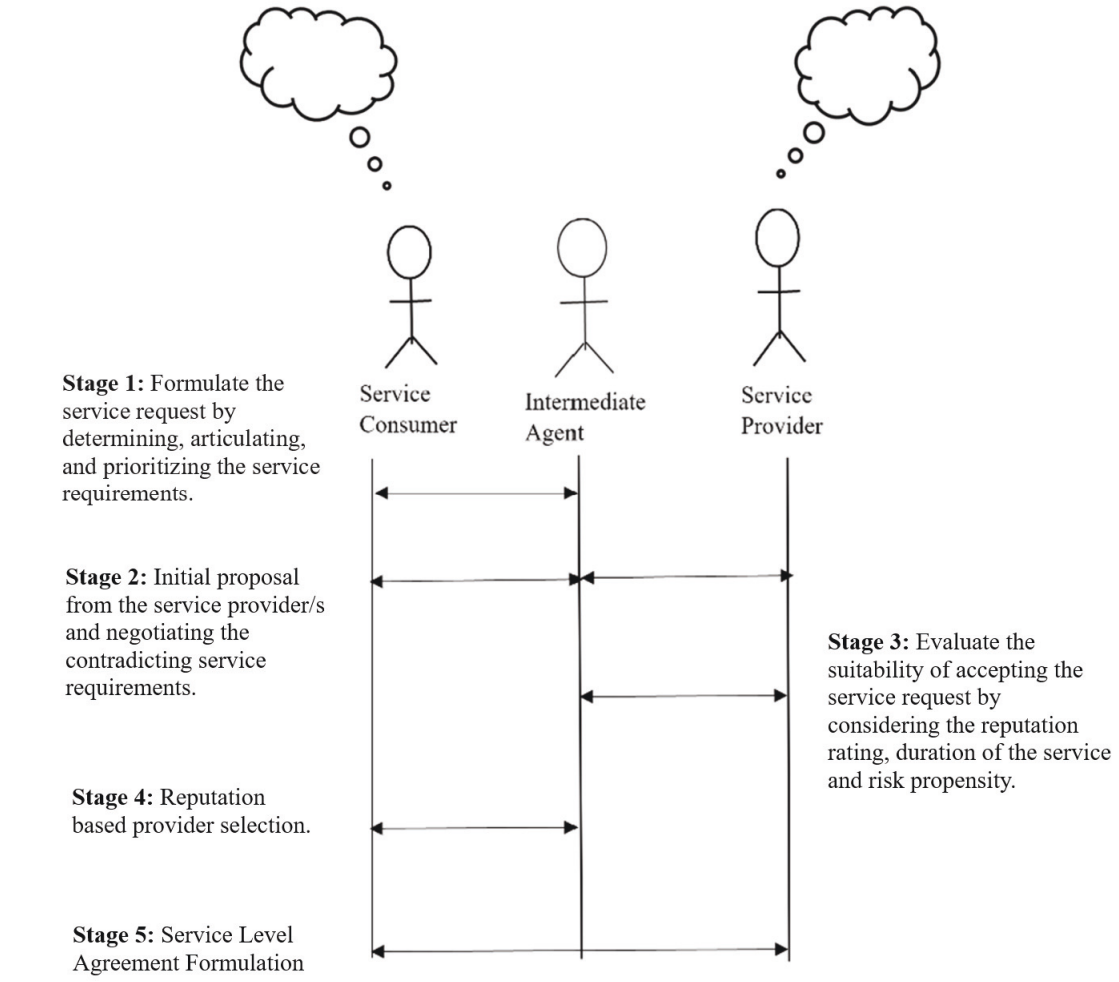


Figure 5.1: *Service Negotiation Framework* by S. Khan, 2022

Stage 4: Reputation-based provider Selection

In this stage, the intermediate agent (registry/broker) assists the service consumer in selecting the most suitable provider from the available ones.

Stage 5: The SLA between the two agents is formulated and executed – In this stage, the customized SLA between the service consumer and the selected service provider is formulated and implemented.

In the following subsections, we explain the working of each stage in detail.

5.4.1 Stage 1: Formulation of a Service Request

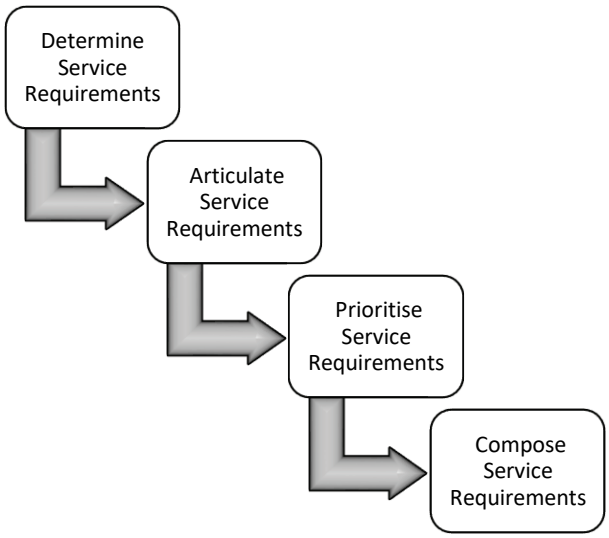


Figure 5.2*: Formulate the Service Request (Pre-Negotiation)* S. Khan, 2022

The *formulation of a service request* is the process of determining, articulating, and prioritizing the service requirements with a quantitative or qualitative or hybrid expression. The service requirements can be expressed as numeric, text, or a combination of numeric and non-numeric measures(Fachrunnisa, 2011). Service request formulation is a vital task. This step plays a pivotal role in contributing to successful SLA management and creating a trustable relationship between service providers and consumers. Before the interaction, if there are difficulties in determining the service requirements or service criteria because these have not been well thought out by the interacting parties and articulated in unambiguous terms, it may be difficult to assume the interacting partner knows what is expected of it. We propose four steps in service request formulation to address the above issue, which leads to the service negotiation process.

In addition, maintaining valuable trust relations can be reflected in delivering services (possibly personalised services) in a unique and valued manner(Hussain, 2006). To ensure personalised service delivery, each interacting party needs to determine their service requirements at a granular level. Delivering personalised services gives the service consumer the impression that the service provider regards them as a priority consumer and cares about their needs.

The service provider will be better able to deliver on the consumer's service requirements if it has a clearly articulated idea of their requirements. We argue that trust and personalised service delivery could be achieved and maintained only by consistently delivering services according to the requirements that have been agreed upon. This can be achieved only if the service requirements have been determined and composed by both parties in a granular manner prior to service delivery.

The steps for service request formulation are as follows:

Step 1: Determine the service requirements

Step 2: Quantify the service requirements

Step 3: Prioritise the service requirements

Step 4: Compose the service requirements

Details of these steps are discussed in the following sections.

5.4.1.1 Step 1: Determine the Service Requirements

In this step, the service consumer, in consultation with the intermediate agent, defines and formalizes the requirements expected of the service. As shown in Table 1, we propose a standard service formulation template that assists the intermediate agent in collecting the service requirements from the service consumer and structuring them accordingly. This step involves completing the service criteria part of the template, which enables the service consumer to define the textual description of the required service and the applications dependent on it. The intermediate agent uses this information to appropriately identify the service provider/s best suited to meeting these requirements. Extending the aforementioned

scenario between ABC Education and SMART Mediation, let us consider that ABC Education needs a computing infrastructure of different types to provide online education to its clients. One of the criteria it requires as part of the service is the ability to provide bulk data transfer to different locations (Zone A and Zone B) with varying priority levels. For example, let us assume that Zone A of ABC Education consists of a software development unit and runs cloud computing servers and Cisco certification programs, the core unit of ABC Education. This zone also provides software support and service to ABC Education's students.

On the other hand, let us assume that Zone B deals with general courses only that do not have as much traffic as Zone A. So, as shown in Table 5.3, the service criteria component of the standard service formulation template enables the service requestor to define the different level/s of differentiated service as required by the same requester for different locations and the different aspects required under each criterion. Using this service formulation template, the business parties explicitly affirm and articulate their service requirements in such a way that they are easily understood or can be interpreted without any assistance. It also avoids any unexpected future confusion among the interacting parties.

Service Criteria			**Quality Descriptors**	**Level of Importance**
Service List	Service requirements description	Application dependency		
S_1			Q_1 ……Q_n	Imp_0
…..			……	…..
….			……	……
S_n			Q_{n1}…….Q_{mN}	Imp_m

Table 5.3: Service Requirement Formulation Templates (Service requester’s statement)

5.4.1.2 Step 2: Quantify the Service Requirements

Quantifying service requirements is the process of expressing the service requirements in terms of service criteria or service needs and their associated quality descriptors. To articulate service requirements, we propose a standard service formulation template that is effective for service consumers and service consumers to collect and structure the service information. Moreover,

by using this service formulation template, the business parties explicitly affirm and articulate their service requirements so that they are easily understood or able to be interpreted without any assistance and avoid any unexpected future confusion between the interaction parties. This template may vary depending on the type of services, QoS level, and parties' interactions. We design this template to deliver other service-oriented components rather than only SDN services. The template details are provided below, and the template elements are a list of service criteria, their quality descriptors, preferences or level of importance. These elements may vary according to business demand.

Once the service criteria and the different aspects under it are defined in the previous step, in this step, those same aspects from that criterion are numerically quantified to have their quality descriptors defined.

5.4.1.3 Step 3: Prioritise Service Requirements

Service requirements prioritisation is the process of evaluating the importance or urgency of each service requirement. We represent this value as four levels:

Less Important – 0

Moderate Important – 1

Important – 2

Highly Important – 3

Let us consider that the service formulation template is constructed and used by both the service requester and the intermediate agent. However, in our case, the initial draft is created by the service requester. A standard service request template is developed and shown below, where the service requester starts drafting their articulated service requirements that they have previously determined. Returning to the aforementioned case, ABC Education prioritises its requirements according to its business requirements. It makes statements about our standard service request templates, containing service criteria, level of importance, and quality descriptors. ABC Education takes some expert advice at this stage; therefore, they interact with

their intermediate agent (SMART Mediation) before composing a final statement. Therefore, the outcome generates valuable opinions from both parties with necessary corrections. As a result, SMART Mediation clearly understands ABC Education's service requirements and expectation level. They were involved from a very early stage of service requirements formulation.

Table 5.3 shows the high-level template of service requirement formalization. In this framework, we express that service requirement formulation is a decisive factor, as service quality and satisfaction depend entirely on the requirements of the requested or provided services. For example, the requester requests a service S1 to receive various services in various zones or areas. Suppose we consider that the R1 requester requests S1 services for A zone and B zone. A zone deals with very critical applications all the time and is used by very high professionals. Therefore, they require the best Q1-level services. As they are critical applications, service availability is crucial, which means this is given a very high ImP3 level of importance. On the other hand, Zone B deals with mid-level applications which are used for educational purposes and business hours availability is critical. Thus, for Zone B, the S1 service quality level can be defined as the Q2 level, and the level of importance can be defined as Imp2.

5.4.1.4 Step 4: Compose Service Requirements

Service requirements composition is the process of arranging the service requirements from the service criteria or service needs and its associated quality description to which both interacting parties have agreed in a predefined structure and generates a document for both parties. To provide a methodology for composing service requirements, we propose a detailed service formulation template for service consumers and intermediate agents.

Let us discuss the above case in more detail to compose the service requirements for their requested services. In the service requirements prioritisation section, we stated that a service could be ordered in a different manner by the same requester. Let us assume that zone A of ABC Education consists of a software development unit and runs core units such as cloud

computing, Cisco certifications, etc. This zone also has software support and a service unit. On the other hand, Zone B deals with general courses only.

Service Criteria: Service list	Service requirements Description	Application dependency	Service Duration	Quality Descriptors	Level of Importance
SR1= **Support Bulk Data** **Service Definition:** This type of traffic includes important email transactions and video and content distribution. Backup operations can be performed after hours	This type of traffic includes important email transactions and video and content distribution. Backup operations can be performed after hours	Email, Backup Operations, FTP/SFTP transfers, Video and content distribution	2 years		
Zone A: Zone Location S1 - Full availability of email services at all time S2 - FTP /SFTP transfer should not have any interruption S3 - Video and content distribution should not interrupt during business hours S4 - Backup Operations should perform after hours.	**Full Availability**: the percentage of the available average service time (AST) can be measured weekly, monthly or quarterly. **Uninterrupted FTP/SFTP**: unencrypted and SSH authenticated method to transfer from server to local systems or to remote server with reliability	Email, Backup Operations, FTP/SFTP transfers, Video and content distribution	2 years	Q1=Email services avg percentage 98.9% monthly Q2=FTP/SFTP transfer reliability 98.6% Q3=CBWFQ 60% Q4=backup start from 7:30pm	Imp_3
Zone B: Zone Location **S1 - Full availability of email services at all time** **S2 - FTP /SFTP transfer should not have any interruption** **S3 - Video and content distribution should not interrupt during business hours** **S4 - Backup Operations should perform after hours.**	**Video and Content distribution**: Guaranteed bandwidth (CBWFQ)=size of the video sessions plus 20 percent. Maximum 75% rule **Backup Operations:** should trigger the operations after 6:30 pm	Email, Backup Operations, FTP/SFTP transfers, Video and content distribution	1 year	Q1= Email services avg percentage 98.0% monthly Q2=FTP/SFTP transfer reliability 98.0% Q3=CBWFQ 50% Q4=backup start from 6:30pm	Imp_2

Table 5.4: Composed service requirements statement.

5.4.2 Stage 2: Proposal from the Service Providers

In this stage, the intermediate agent takes the service request which was formalized in the previous stage and solicits responses from potential service providers. Before doing this, it determines and addresses any ambiguous or contradictory requirements present in the

formalized service request to ensure that the consumers' needs are understood in a differentiated manner. The aim of this stage is achieved in three steps as follows:

5.4.2.1 Step 1: Translate the Service Request to a Service Description Template (SDT):

In this step, both parties review and discuss their service requirements. In order to maintain a trusting relationship, negotiation is a key element. The term negotiation is described as a decision-making process; thus, the parties exchange information, initiate comprehensive proposals, establish proposals and advice, deal with independent tasks, and search for an arrangement based on cooperation and reconciliation decision-making (Fachrunnisa, 2016). In the same context, we assume that the service requirement is described from the perspective of both the service requester and the intermediate agent. As a result, if, from the intermediate agent's point of view, the proposed service requirements are challenging to achieve or cannot be delivered to the service requester or vice versa, an adjustment of the proposed service requirements will take place to accommodate the interest of both parties. To compare and translate the service requirements (Table 5.5), a service description template (SDT) is proposed (Fachrunnisa, 2016). Therefore, the intermediate agent translates the formulated service request into an SDT (Fachrunnisa, 2016). The SDT, as shown in Table 5.5, compares aspects of the service requirements from the perspective of the service requester and the intermediate agent.

Service Request ID:				
Service Requester ID:				
Service Provider ID:				
Service Time Frame: **Service Time Slot:**				
Number of Checkpoints				
Service Requirements Descriptors	**Service Requester's Proposal**		**Service Provider's response**	
	Service Quality Descriptors	Priority/Level of Importance	Service Quality Descriptors	Priority/Level of Importance

Table 5.5: Service Description Template (SDT)

5.4.2.2 Step 2: Identify and Address Any Conflicting or Unfeasible Requirements:

After the SDT is formed and if, from the perspective of the intermediate agent, the proposed service requirements are either *unfeasible* or difficult to achieve, an adjustment to the service requirements will need to take place to accommodate the interest of both parties. This is done in this step through a process of negotiation to address any conflicting requirement/s. Negotiation is described as a decision-making process in which the involved parties exchange information with comprehensive proposals, identify aspects that are unclear or cannot be committed to and search for an arrangement from a cooperation and reconciliation decision-making point of view (Fachrunnisa, 2016). At the end of this negotiation process, the SDT, defined in the previous step, is updated with the requirements that were changed due to them being identified as conflicting.

Service Request ID: SRI00056						
Service Requester ID: SR0006						
Service Provider ID: SP0164						
Service Time Frame: 1 January 2021 –30 January 2024 **Service Time Slot: Every 6 hours**						
Number of Checkpoints: 2						
Service Requirements Descriptors	**Service Requester's Proposal**			**Service Provider's response**		
SR1 = Support Bulk Data	Service Quality Descriptors	Priority		Service Quality Descriptors	Priority	
		Zone-A	Zone-B		Zone - A	Zone- B
S1- Full availability of email services all time	Q1= Email services avg percentage 98.9% /month	Imp3	Imp2	Q1= Email services avg percentage 98.1%/month	Imp3	Imp 2
S2- FTP /SFTP transfer should not have any interruption	Q2=FTP/SFTP transfer reliability 98.6%	Imp3	Imp2	Q2=FTP/SFTP transfer reliability 96.6%	Imp3	Imp 2
S3- Video and content distribution should not interrupt during business hours	Q3=CBWFQ 60%	Imp3	Imp2	Q3=CBWFQ 55%	Imp3	Imp 2
S4- Backup Operations should perform after hours.	Q4=backup start from 8:30pm	Imp3	Imp2	Q4=backup start from 7:30pm	Imp3	Imp 2

Table 5.6: Formulation of the Service Description Template (SDT)

5.4.2.3 Step 3: Solicit Proposals from the Prospective Service Providers and Address any Ambiguous Responses, if Present:

In this step, the intermediate agent plays the role of a broker and solicits proposals from potential SDN service providers who can satisfy the consumers' needs as defined in the SDN. As a result, the intermediate agent receives *offers* from different service provider/s who have the capability and are willing to provide the infrastructure to satisfy the requirements. An offer from a service provider consists of the quality descriptors it can provide in response to each aspect of the requirement after evaluating the suitability of accepting the particular request. We discuss in the next section the process of a service provider evaluating the suitability of accepting a request. However, as in the previous step, some quality descriptor responses from a service provider in terms of what it can commit to may be ambiguous. In this step, the intermediate agent's job is to identify and clarify such ambiguous responses with them to address any concerns. After this is done, the intermediate agent updates the SDM with the responses received from the different service providers.

5.4.3 Stage 3: Assisting the Service Providers in Evaluating the Suitability of Accepting a Service Request

In this step, we present a framework for evaluating the suitability of accepting the service request. Let us consider that service providers and service requesters are discrete agents. According to the previous discussion, we already know that the intermediate agent is an individual and unbiased agent. They will be used to administer the interaction between the service requester and service provider during the progression of interaction at any point in time. Hence, the objective of this interaction is as follows: a) to provide uninterrupted services with guaranteed service quality; b) to deliver differentiated and personalised services to the service consumer; c) to develop and maintain a standard level of trust between service consumers and service providers (the process of selecting the service provider is the fourth step of our proposed methodology).

Let us consider the aforementioned case where ABC Education is looking for services. For this purpose, the requester has been deriving numerous interactions with the intermediate agent SMART Solutions. These interactions assist in developing a selection criteria statement upon both parties' understanding and demands. A further modified and comprehensive selection criteria statement is developed to find and avoid any possible contradiction between a requester and an intermediate agent. As a result, a final version of the service requirements statement is documented with 'm' service criteria mutually agreed upon by ABC Education and SMART solutions. Therefore, SMART solutions distribute the service request statement among the service providers. In this stage, the service providers evaluate their suitability to accept the service request. Let us assume that Xquery Ltd., an SDN service provider, is one of the interested service providers who would like to deliver the service request.

Let us consider that Xquery Ltd. is a well-established service provider company that has delivered SDN services for the last seven years. Xquery Ltd. has a good reputation in the current provider market. Xquery Ltd. has an apparent understanding of its business scope and limitation. However, providing network services involves utilizing company resources such as hardware resources, software resources, expertise, etc. As a result, it is important for a service provider company to and has a specific number of resources to deliver the service request considering their current circumstances. To make this decision, the service provider evaluates the service request and makes their decision

At this stage, we developed a framework to evaluate the suitability of accepting the service request and assist in the service provider's decision-making. To make this intelligent decision and assess the suitability of the provider to deliver the requested services, the following factors are considered

a) Factor 1: Reliability of the consumer

b) Factor 2: Duration of the services

c) Factor 3: Service risk propensity

5.4.3.1 Reputation Rating Scale

The generic concept of reputation drives the specific concepts of "*service reputation*" and "*product reputation*". The term "*service reputation"* is defined as aggregating all third-party recommendations from agents' recommendations(Hussain, 2006). The recommendation is considered as the reply or feedback to a service requestor's reputation query about the QoS being delivered by a given service provider in a given time slot. In the same way, "*product reputation*" is the collection of recommendations from all third-party recommendation agents in response to a buyer's reputation query about the product in a given context and a given time slot.

In order to qualify the reputation of an entity that could validate the framework quantitative and numerically, we propose a reputation rating scale in this thesis. The inspiration behind employing a reputation rating scale is that a certain entity may have a different reputation rating scale depending on its behaviour with those entities who have worked together and had an interaction. Therefore, the reputation or the feedback can be computed. Thus, we propose the concept of a reputation rating scale to compute and express the reputation of an entity. Hence, we define the *reputation levels scale* as a numeric quantity that portrays the level of reputation of an entity.

We propose six different reputation levels in the range of [0-5]. The values 0 to 5 represent an ordinal scale. We also define the semantics of each reputation level so that once a given reputation querying agent has determined a given reputation level, it can understand the meaning of the computed reputation level. The semantics are shown in Table 5.7. The domain of the different possible reputation levels is shown in set R.

$$R = (0, 1, 2, 3, 4, 5)$$

We choose the above domain to represent the reputation values to reflect the fact that there is a close relationship between the concepts of trust and reputation. The reputation of an entity is determined solely by recommendations from third-party recommendation agents. The recommendations are the trustworthiness value assigned by the third-party recommendation agent to the reputation-queried entity when they interact with each other. To maintain

consistency, there has to be a one-to-one correspondence between each level of trust and reputation. Additionally, the semantics of the levels of trustworthiness and reputation have to agree with each other. It should be noted that when we refer to the term *reputation*, we mean the reputation of an entity in a given context and time slot.

Reputation Rating Scale	Semantics
1	Extremely bad reputation
2	Bad reputation
3	Partially good reputation
4	Good reputation
5	Very good reputation
0	No Rating

Table 5.7: Reputation rating scale

A reputation level of 0 denotes that the reputation querying agent could not ascertain or determine the reputation of a given reputation-queried entity. This may be because it obtained no recommendations from the third-party recommendations or the obtained recommendations were from untrustworthy known agents.

Reputation levels 1 and 2 denote a negative reputation (or disrepute). We define disrepute as a pessimistic, cynical, disparaging, adverse or unfavourable assessment by the reputation-querying agent about a given reputation-queried entity due to soliciting recommendations about it from the third-party recommendation agent(s). Reputation level 5 denotes the highest level of disrepute, and reputation level 1 is used to denote the lowest level of disrepute.

The repute levels of 3 and 4 denote a neutral reputation. We define a neutral reputation as a mediocre or average assessment by the reputation-querying agent about a given reputation-queried entity as a result of soliciting recommendations from the third-party recommendation agent(s). Repute level 3 denotes the lowest level of neutral reputation, and repute level 4 denotes the highest level of neutral reputation.

The repute levels of 4 and 5 denote a positive reputation. We define a positive reputation as an affirmative or sanguine assessment by the reputation-querying agent about a given reputation-queried entity due to soliciting recommendations about it from the third-party recommendation agent(s). Repute level 4 denotes the lowest level of positive reputation, and repute level 5 denotes the highest level of positive reputation.

5.4.3.2 Factor 1: Reliability of the Service Consumer

The *reliability* of the service consumer is an indication of the nature of the past relationships between itself and the contracted service providers. In other words, the reliability value of a consumer represents its summarised service transaction profile which is the opinion of the service providers who have interacted with it in the past. This transaction profile not only consists of the detailed history of the service user's every transaction but also contains a reputation rating provided by all of the contracted service providers to the service consumer. A high reputation rating indicates that the service providers rate the consumer highly in delivering what it promised, ease of dealing with it, etc. Thus, they have a high level of trust and willingness to deal with it again. This is seen as a positive factor by prospective service providers when deciding whether to accept a request from a particular service consumer or not.

Similarly is the case vice versa. The intermediate agent stores all the past transactions of a service requestor with different providers, along with the reliability value given by each of them. A prospective service provider uses that history to ascertain the reliability value of a service consumer in a future transaction.

While calculating the reliability of a service consumer, its previous 'n' years of reputational data in (a) the consumer's overall reputation rating *(R)* and (b) the service provider's satisfaction rating to this provider (OPR) are considered. Then, dividing them by their maximum reputation rating. The reliability calculation performed using calculation function (∫) where the ∫ = (+) or (-) or (*) or (/). The equations are given below.

$$\boldsymbol{R = \int(Reputation\ rating, Other\ Providers\ Reputation\ rating)} \quad \textbf{Equation 1}$$

$$\boldsymbol{R = \int \left(\frac{Reputation\ rating, Other\ Providers\ Reputation\ rating}{Max\ Reputation\ rating, Max\ Other\ Reputation\ rating} \right)}$$ **Equation 2**

$$\boldsymbol{R = \int \left(\frac{R\ rating, OPR\ rating}{Max\ R\ rating, Max\ OPR\ rating} \right)}$$ **Equation 3**

$$R = 5 \times \left(\left(\frac{R\ rating}{Max\ R\ rating} \right) \left(\frac{OPR\ rating}{Max\ OPR\ rating} \right) \right)$$ **Equation 4**

In this equation, the reputation rating range is from 1 to 5, where 1 represents the lowest and 5 represents the highest. The Max R rating is the maximum score within the range that the ABC Education company can receive from its consumers, which is 5. The Max OPR rating is also determined similarly. We scale the rating in the following orientation (Table 5.8). Therefore, we determined that the R-value of each service consumer is then represented by a fuzzy membership value, as illustrated in Figure 5.3.

R-value scale	Description of the scale
R=0 to R=1.5	Less Reliable
R= 1.5 to R= 3.5	Reliable
R = 3.5 to R= 5	Highly Reliable

Table 5.8: R-value scale description

We use the Mamdani method to calculate the provider's suitability to assist in critical, intelligent decision-making. We used MATLAB to articulate the input variables of the membership functions. Figure 5.3 represents the membership function plot of the input variable "Consumer company reliability" (left). The company's reliability is articulated in three categories: Less Reliable, Reliable, and Highly Reliable.

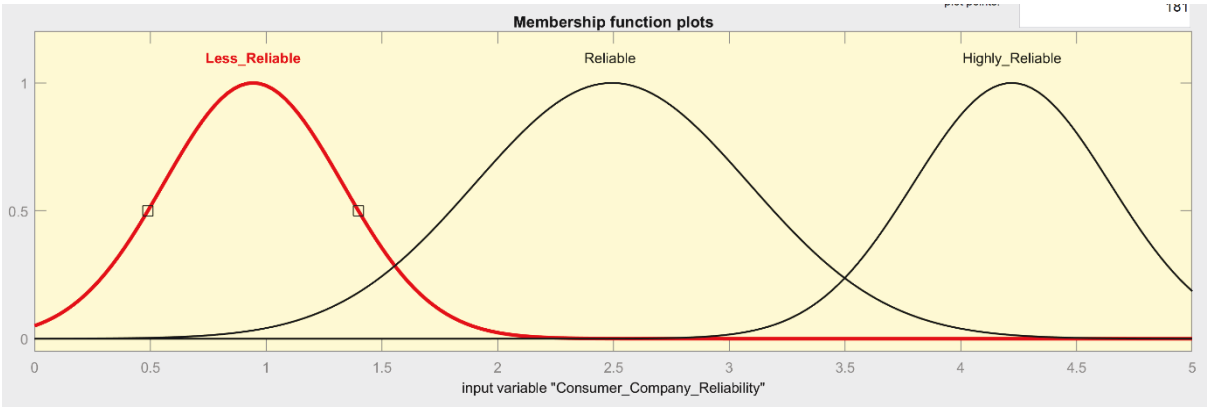


Figure 5.3: Membership function plot of the input variable "Consumer Company reliability"

5.4.3.3 Factor 2: Duration for Which the Resources are Requested

The duration for which the resources are requested is the second factor considered by the service provider when ascertaining the suitability of accepting a request or not. It depends on the service provider as to how it views this criterion. For example, some providers may prefer to form a long-term agreement with a service requestor to ensure it can commit to using its resources. On the other hand, some providers may already have committed to using their resources at a future time period, t_{m+p}, and may have the required resources available only from the current period of time t_m till t_{m+p}. So this criterion plays an important role in determining if a request from a service provider should be accepted or not. As shown in Table 5.9, the proposed approach uses a rating range to represent the duration for which the consumer requests the resources. This rating is used further when determining the suitability of accepting a service request. The duration for which the resources are requested is represented on a fuzzy membership value, as shown in Figure 5.4.

Duration	Description of the duration	Rating
0 to 3months	Service required from 0 Months to 3 Months	Too Short
0 m to 6 m	Service required from 0 Months to 6 Months	Short
0m to 12 m	Service required from 0 Months to 12 Months	Medium
0m to 18m	Service required from 0 Months to 18 Months	Standard Medium

0m to 24m or more	Service required from 0 Months to 24 Months or more	Long

Table 5.9: Service Duration Rating Scale

Furthermore, the provider will decide to accept or reject the service request in light of the service duration rating scale. According to the scale, a request for 0 to 3 months duration is less suitable from the service provider's point of view as this will be only a short-term contract. On the other hand, a service request of more than ten years duration is considered to be more reliable and suitable for the service provider. A longer-duration service request is preferable and is considered to be a high priority.

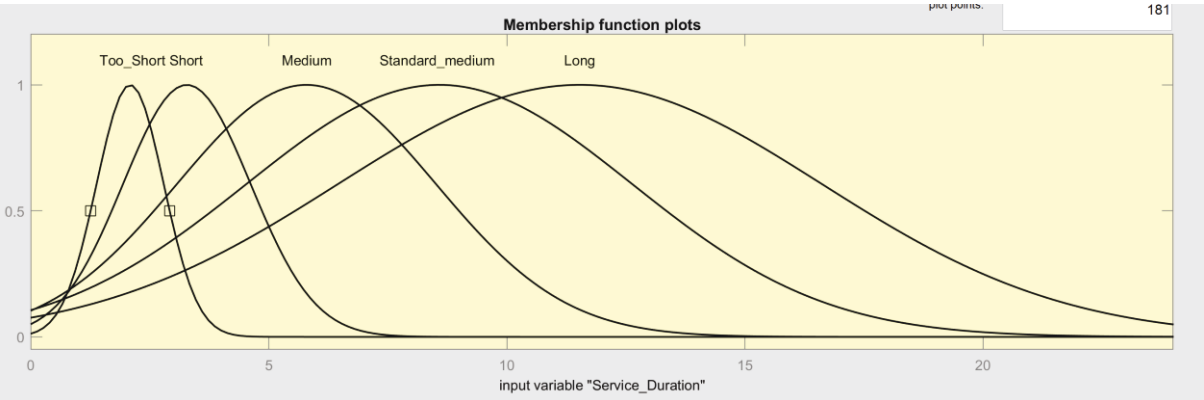


Figure 5.4: Membership function plot of the input variable "Service Duration"

5.4.3.4 Factor 3: Risk Exposure of the Service Provider in Accepting a Service Request With the Required Specifications

Risk exposure is an important factor to consider when making an intelligent decision about a service provider's suitability. This criterion determines the expenses to be incurred and ascertains its risk using the risk scale. For example, continuing with the above scenario, if provider Xquery Ltd. is planning to deliver the requested services to the requestor ABC Education, then from the perspective of Xquery Ltd., different cases may arise, such as,

Xquery Ltd. requires x amount of services for the following year. After receiving the service requirement statement with 'm' service criteria from the requester company (ABC Education), Xquery Ltd. assesses the service requirements criteria. This assessment includes a detailed investigation of the following discussion items:

- It does not have sufficient hardware and software resources to provide x amount of services to ABC Education.
- It has adequate hardware and software resources to deliver x amount of services; however, it has a deficient technically skilled staff to provide x amount of services at the required level. In this case, they need to hire expertise.
- It has adequate hardware, software, and technical expertise; however, the available resources are not adequate to deliver the x amount of services requested for the whole year.

Considering the above cases, the service provider company calculates the risk exposure if it purchases additional infrastructure or resources for the next year to deliver the requested services. To evaluate the possible risk exposure of a provider, we evaluate the provider's current risk exposure (Table 5.10) if the provider company agrees to deliver the service request and then divide this by the Maximum Risk exposure followed by performing a calculation function (∫) where ∫ = (+) or (-) or (*) or (/). The risk exposure of the service provider in accepting service with the required specifications is represented as a fuzzy membership value, as defined in Figure 5.5.

$$\textbf{Risk R} = \int \left(\frac{\boldsymbol{Current\ Risk\ Level}}{\boldsymbol{Maximum\ Risk}} \right) \qquad \textbf{Equation 5}$$

Risk Rating	Resource Used	Expense Range
Low Risk	No additional resource requirements to 20% of their resources required	$0.00 - $n1

Medium Risk	20% of their resource require to 50% of their resources required	$n1 - $n2
Risk	50% of their resource require to 80% of their resources required	$n2 - $n3
High Risk	80% of their resource requirements to 100% of their resources required	$ n3 - $n4

Table 5.10: Possible Risk Rating: The following risk rating table depicts the maximum risk a provider company may take, rated as five, and the resources used and the expenses range is changeable.

Additionally, in the other membership function plots, the service risk input variable is divided into four categories: Low Risk, Medium Risk, Risk, and High Risk, which are shown in Figure 5.5.

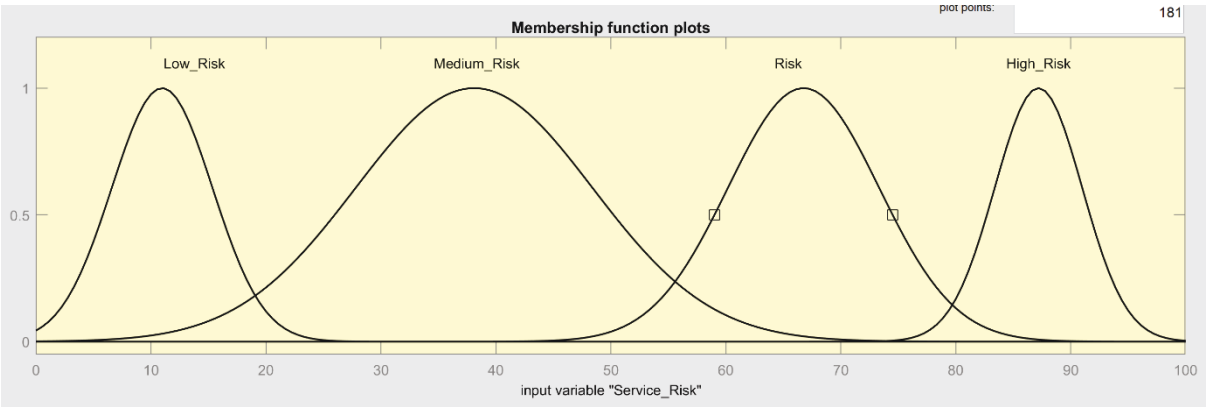


Figure 5.5: Membership function plot of the input variable "Risk Exposure"

5.4.3.5 Suitability Calculation:

In this step, the suitability for the provider to accept the service request from the consumer is determined using Equation 6. We use the "If-Then" rules to receive the output. According to the inputs, 60 variation rules are constructed (a screenshot is provided below). The duration for which the resources are requested is represented on a fuzzy membership value, as shown in Figure 5.6, and the resulting scale is shown in table 5.11.

$$Suitability = Reliability \ \&| \ Duration \ \&| \ Risk \ Propensity \qquad \textbf{Equation 6}$$

Description of the scale	Result Scale
Less Suitable	1 to 20
Moderately Suitable	20 to 40
Suitable	40 to 80
Highly Suitable	80 to 100

Table 5.11: Scale of the Result

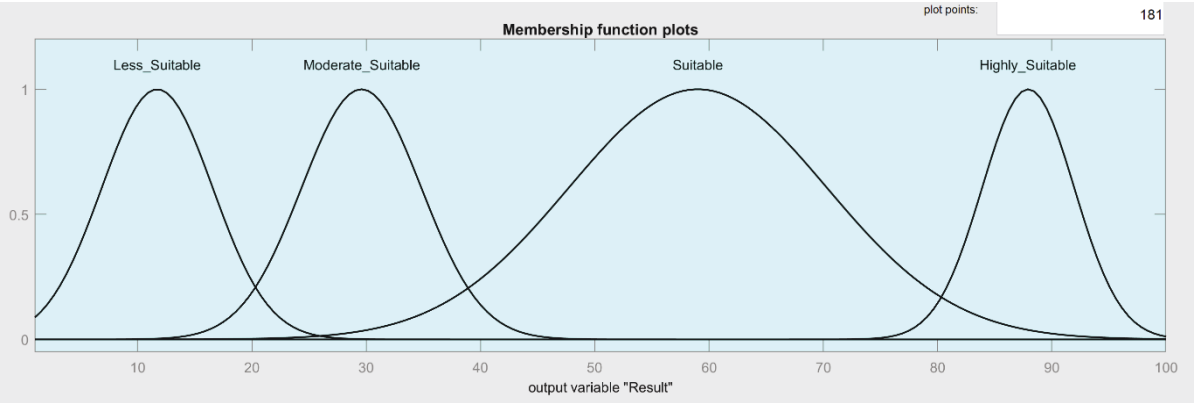


Figure 5.6: Membership function plot of the output variable "Suitability"

5.4.4 Stage 4: Assisting the Service Requestor in Selecting the Most Suitable Service provider

We developed a framework for the requester or consumer company to select a suitable service provider. According to our service-oriented-based framework, we know that service providers and service requesters are discrete agents. We also know that the intermediate agent is an individual and unbiased agent. They will be used to administer the interaction between the service requester and service provider during the progression of the interaction at any point in time. Hence, this interaction aims to develop and maintain a level of trust between service consumers and providers. Selecting the service provider is the fourth step of our proposed methodology.

The framework we develop is principally reputation data-driven provider selection. This framework leverages the opportunity for the requester or consumer organization to assist in suitable provider selection. Let us assume that a service requester company, ABC Education, is looking for network services for their organization. For this purpose, the requester has been engaged in numerous interactions with the intermediate agent SMART Solutions. These interactions assist in developing a selection criteria statement based on the understanding of both parties concerning the demand. A further modified and comprehensive selection criteria statement is developed to find and avoid any possible contradiction points between the requester and intermediate agent. As a result, a final version of the service requirements statement is documented with 'm' service criteria mutually agreed upon by ABC Education and SMART Solutions. Therefore, SMART Solutions distributes the service request statement to the service providers. Service providers evaluate their suitability using our proposed suitability framework (Step 3) and either accept or reject the service request. Let us assume that an "n "number of service providers are very interested in delivering the services and evaluating that ABC Education's service request is a reasonable service request for them.

According to the previous discussion, we already know that this interaction's objective is a) to provide uninterrupted services with guaranteed service quality, b) to deliver differentiated and personalised services to the service consumer, c) to develop and maintain a standard level of trust between service consumers and service providers. The process of selecting the service provider is the fourth step of our proposed methodology. In this stage, the consumer has the opportunity to select a suitable service provider among the 'n' number of interested providers

to whom they can rely on and feel confident to make a contract. In other words, the requester company needs to find a reliable service provider to deliver their requested services.

In this stage, the intermediate agent assists the service requestor in selecting the most suitable service provider from among those who responded with an offer. It does this by determining the suitability of each service provider by showing how closely associated each provider is with the requestor's needs. The service providers who responded can broadly be divided into established and emerging categories. Established providers are those who have been providing their services for more than two years. In other words, these providers are those who have a reputation value given to them by the previous service requestors.

On the other hand, emerging providers are either new or have not been providing services for more than two years. Compared to their established counterparts, these providers may not have a high reputation value and thus may be disadvantaged when the service requestor decides between this agent and the established ones. To address this, the process by which the intermediate agent ascertains the suitability value for each provider varies, as explained in the following sub-sections.

5.4.4.1 Determining the Suitability Value of an Established Provider

As mentioned earlier, established SDN providers have been in the market for more than two years. In other words, these providers have a reputation value from the previous service requestors. To determine the suitability value for the current transaction, the service requestor considers two main parameters, namely (a) how many transactions have been performed by the provider so far (T_n) and (b) how many of these transactions were marked as successful by the service requestor (T_s). The transaction trend of the provider can be determined using the following equation (Equation 7), and the suitability of the service provider can be determined using equation 11.

$$T_s/_{Tn} \times 100 \geq Threshold \qquad \textbf{Equation 7}$$

If the suitability value is equal to or exceeds the predetermined threshold, then the consumer may consider the provider as a suitable provider. The consumer can determine the acceptable percentage of success ratio. This acceptable success ratio can vary with various requester company's demands.

We used the Mamdani method to determine a suitable service provider for the requested services and to assist in intelligent decision-making in relation to provider selection from a range of existing service providers. We used MATLAB to articulate the input variables of the membership functions and the output variables. Figure 5.7 represent the membership function plot of the input variable "Reputation_rating", where the company reputation is divided into three categories: Low, Medium, and High. On the other hand, in the other membership function plots, the "Transaction_trend" input variable is divided into three categories: Decreasing, Standard, and Increasing, are presented in Figure 5.8.

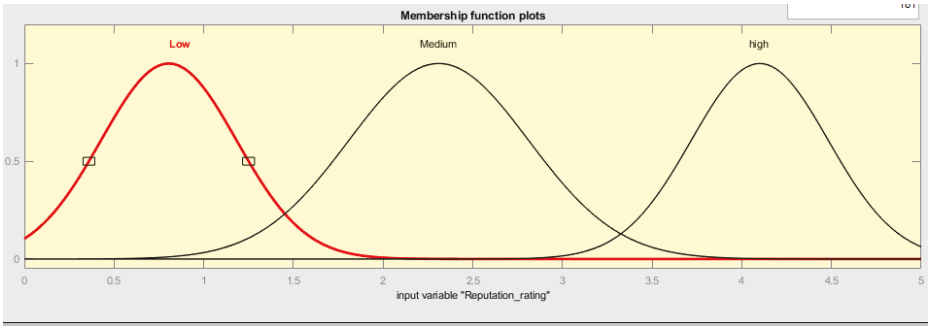


Figure 5.7: Membership function plot of the Input Variable "Reputation_rating"

Table 5.12 details the input variable's membership function plot scale "Reputation_Rating" and "Transaction_trend."

Scale	**Reputation Rating**	**Scale**	**Transaction trend**
0 to 1.5	Low	**0 to 35**	Decreasing

1.5 to 3.5	Medium	**35 to 70**	Standard
3.5 to 5	High	**70 to 100**	Increasing

Table 5.12: Reputation Rating and Transaction Trend scale

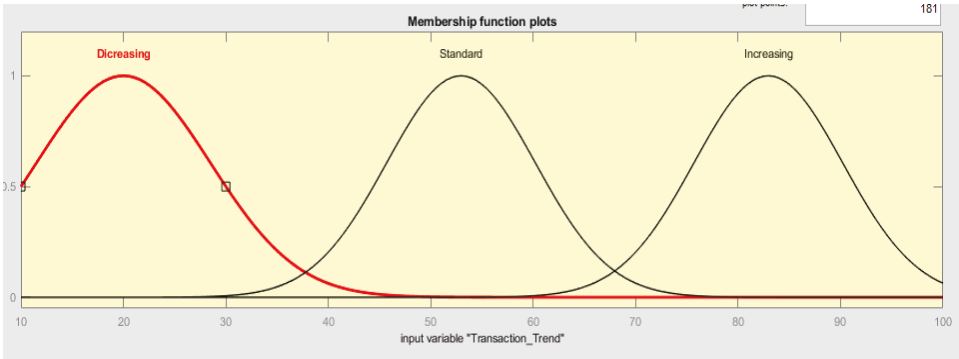


Figure 5.8: Membership function plot of the Input Variable "Transaction_Trend"

Figure 5.9 below illustrates the membership function plot of the output variable "Accept/Reject_Decision", where provider suitability is divided into two categories: Reject and Accept. Table 5.13 below details the membership function scale of the output variables.

Scale	**Membership function of the output variable**
0 to 45 or 35	Reject decision
35 or 45 to 100	Accept decision

Table 5.13: Output variable scale

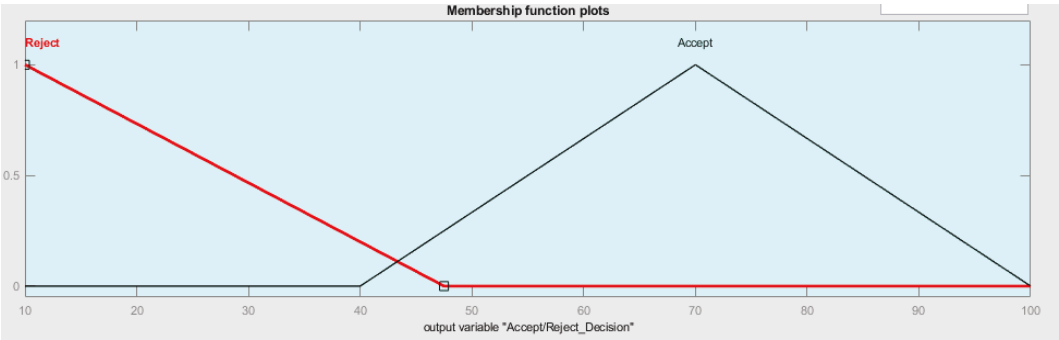


Figure 5.9: Membership function plot of the Output Variable "Accept/Reject_Decision"

We used the "If-Then" method to receive the output. The results of our method are discussed in the results section.

5.4.4.2 Determining the Suitability Value of an Emerging Provider:

As mentioned earlier, SDN service providers who are relatively new to the market do not have previous transaction history that may assist in analysing their suitability value. Therefore, we utilize the k-nearest neighbour (KNN) technique that classifies these agents based on their similarity to existing ones and uses their reputation value to infer their emerging provider's value. The KNN algorithm is a classification algorithm that identifies the majority between the k-most similar instances to a given 'unseen' observation. Euclidean distance is used to measure the k-nearest neighbour's profile to make the predictive decision. We select the top-K nearest neighbours from a set of providers with the maximum number of similarities to a new provider, which is used to make a predictive decision to select a relatively new provider with no previous transaction history. The top-K nearest neighbours comprise three sub-steps:

e) Similar neighbours discovery
f) Define the best k values
g) Transaction trend of the similar neighbours
h) The decision to approve or reject a new provider

K-Nearest Neighbour Algorithm:

K-nearest neighbour or the KNN algorithm is a classification algorithm that identifies the majority between the K's most similar instances to a given 'unseen' observation. In this algorithm, the similarity is demarcated according to a distance matrix between two data points. A popular Euclidean distance method is used in this framework.

The simple Euclidean distance calculation is as follows:

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

The consumer will use the top-K nearest neighbour profile patterns and will make a predictive decision. This method is also used to make a predictive decision to select a relatively new provider who does not have any previous transaction history. We select the top-K nearest neighbours from a set of providers with the maximum number of similarities to a new provider.

a) Similar Neighbours Discovery:

This is the first step in selecting a suitable service provider decision-making process when the service provider is relatively new. At this stage, we find the nearest similar neighbours of a new provider to infer the possible performance of a new service provider. For this approach, we consider some parameters or features to find the nearest features.

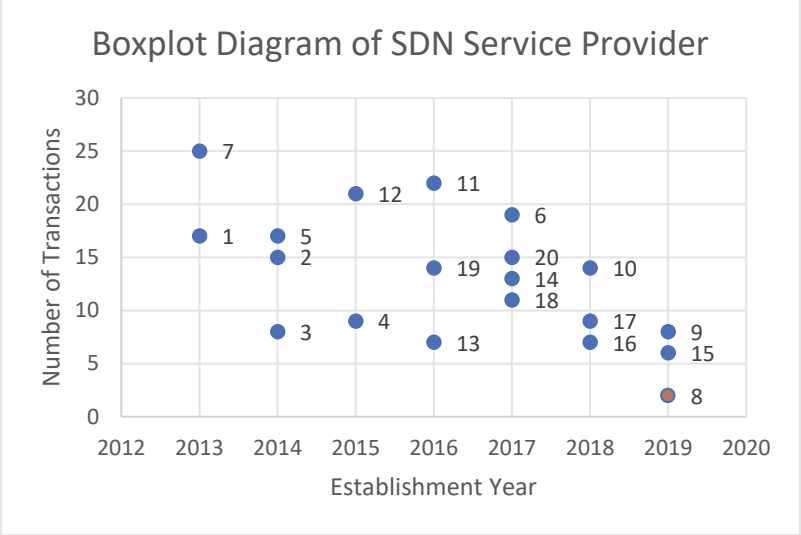


Figure 5.10: Boxplot diagram of SDN service provider

We used the default distance calculation, the Euclidean distance method, widely used for the k-nearest neighbour approach. Let us discuss the scenario based on our dataset, which is discussed later in this section. We represent a boxplot diagram(Figure 5.10) from our dataset where we assume that provider number 8 is a new provider with no reputation rating information. The nearest neighbour (NN) is determined by considering the target provider (provider number 8) r using the following equation.

$$NN=\sqrt{\sum_{i=1}^{k}[(xi-yi)^2+(xa-ya)^2\{yi\in N|ya\in N\}]} \qquad \textbf{Equation 8}$$

where NNs are the nearest neighbours of the target provider,

Xi is the first considered parameter of the target provider,

Yi is the first considered parameter of all neighbours N,

Xa is the second considered parameter of the target provider,

Ya is the second considered parameter of all neighbours N.

The NN values determine the strength of the similarity between a new provider or target provider and existing providers. We used a boxplot diagram to represent the above Euclidian Distance equation result in the results and discussion section. After determining the similarity among these providers, we calculate the strength of the similarity of the target (new) provider. We used the following equation to determine the strength of the similarity.

Sim (r,ra) = $rank(NNr - NNra), Asecnding$ **Equation 9**

In the above equation, sim (r,ra) represents the strength of the similarity between requesting provider NNr and all of its neighbours NNra. Here r represents the requesting provider, and ra is a set of nearest neighbours.

b) Define the best K values

K-values in the KNN algorithm is a parameter that refers to the number of nearest neighbours that gives the best outcome, which is included in this research. The optimal k value can be determined by calculating the square root of N, where N is the total number of samples. The most used technique to find the optimal k value is to derive a plot between the error rate and K-denoting values in a predefined range. Therefore, we choose the K values as having a minimum error rate.

This research used the accuracy plot to find the most favourable K value. The K value varies based on the sample set which is used. For this research, the best K value we obtained is 10.

c) Transaction Trend of similar neighbours

After determining the best K values, we determined how many nearest providers we considered for this research. For this research, we obtained the best K value of 5. Therefore, we calculate the transaction trend for these nearest providers using equation 7.

d) Decision to approve or reject new provider

The consumer approves or rejects a new provider based on their transaction trend. The transaction trend of the new provider is calculated as the mean value of the K-nearest neighbour's transaction trend value. This can be calculated using the following equation.

$$T_{trend\ (rn)} = \frac{\sum_{i=1} T_{trend\ 1} + T_{trend\ 2} + T_{trend\ 3} + \cdots\cdots\cdots + T_{trend\ i}}{i} \qquad \textbf{Equation 10}$$

Using the equation above, the consumer receives an overall understanding of the possible transaction trend of the new provider. At this stage, the consumer re-clarifies their accepted transaction trend threshold, defined in section 7.1.1. Therefore, the consumer evaluates the new provider's transaction trend against the predefined threshold and decides whether to select the new provider as their suitable service provider for the specific service request. For this purpose, the consumer evaluates and makes a decision by considering the following equation (11) below.

Accept request = $T_{trend}\ (rn) \geq threshold$ **Equation 11**

We demonstrate the evaluation result using the fuzzy inference system (FIS) with two parameters, reputation rating and transaction trend, and the outcome is represented in the results section.

Furthermore, the transaction trend of the providers is calculated using equation (12). If provider x has performed Ts number of successful transactions from Tn number of total transactions, we calculate the provider's transaction trend using the following equation

$$Transaction\ Trend_x = \frac{T_s}{T_n} \times 100 \qquad \textbf{Equation 12}$$

We used the Mamdani method to decide whether to accept or reject the service provider for the requested services to assist in intelligent decision-making as provider selection from a range of existing service providers. We used MATLAB to articulate the input variables of the membership functions and the output variables.

For the transaction trend membership function, we used Table 5.12 above for scaling the membership functions scaling of the transaction trend, and Figure 5.8 shows the membership function variable design for the transaction trend. In addition, in the membership function plots, the "Transaction_trend" (Figure 5.8) input variable is divided into three categories: Decreasing, Standard, and Increasing.

On the other hand, we define the input variable “Threshold” with three membership functions. The membership functions are scaled and shown in Table 5.14; the input variable(“Threshold”) design is shown in figure 5.11.

Membership Function	Scale
< Threshold	10 to 35
= Threshold	35 to 70
> Threshold	70 to 100

Table 5.14: Scaling of the input variable “Threshold”

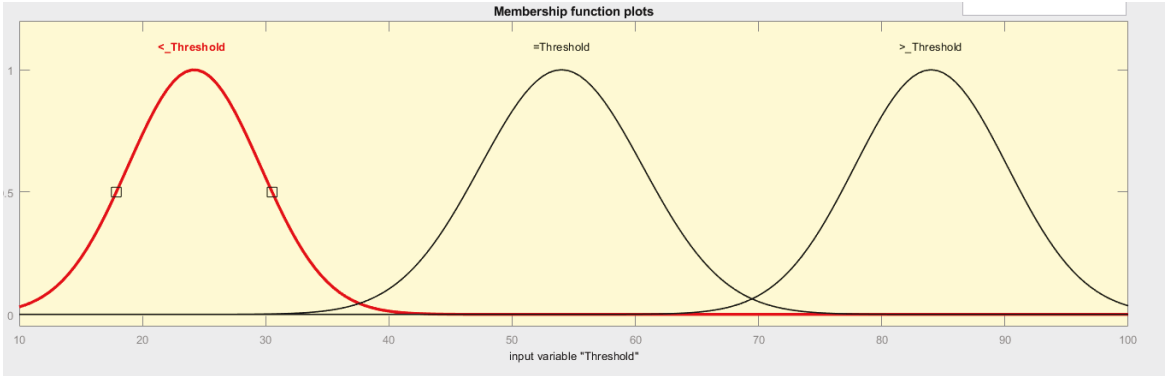


Figure 5.11: Membership function plot of the Input Variable "Threshold"

Furthermore, Figure 5.9 shows the membership function plot of the output variable "Accept/Reject_Decision", where the provider suitability is divided into two categories: Reject and Accept. Table 5.13 represents the membership function scale of the output variables. The details of the results are given in the experiment and result section.

5.4.5 Stage 5: Service-Level Agreement Formulation

SLA formulation is the final stage of the proposed service negotiation framework. The service provider and consumer reach an agreement and make a formal commitment. The service provider already has a clear understanding of the service requester's company from their company profile. The requester company also has a comprehensive knowledge of the provider company. In this stage, the requester produces a formal document that includes a detailed definition of the services that they require, in other words, defining the QoS requirements. This document allows the requester to define their service requirements with finely granulated specifications. It indicates a differentiated service request if required (services for different applications with various requirements) with their expected standard for each service. After reviewing the service request draft, the provider obtains a comprehensive understanding of the requested services. The provider can achieve a personalised service delivery feature by obtaining the requester company's background and service requirements and expectations demarcated in the document. Furthermore, the service provider and requester company negotiate several other business objectives, such as service cost, SLA violation penalties, etc.,

followed by both parties reaching an agreement. The SLA is finalized, formulated, and signed by both parties in the presence of a third party.

5.5 Experimental Validation and Result Discussion

The previous section presents the service negotiation framework for personalised service delivery, which is the first step of our proposed SLA based on guaranteed QoS delivery in an SOA-oriented SDN. As previously mentioned, the negotiation framework is the foundation of a trust-building relationship between the service consumer and the service provider. To determine the effectiveness of our proposed methodology for personalizing the service request and delivering the QoS according to the requirements, we engineered the service negotiation framework. As the framework consists of multiple sub-frameworks, we engineered the framework by making use of the following tools.

1. MATLAB is a proprietary multi-paradigm programming language and numeric computing environment that allows various matrix manipulations, plotting of functions and data, implementation of algorithms, and many more. The environment facilitates various toolboxes to design and simulate, such as Beep Learning and the Fuzzy Logic toolbox. We have extensively used the fuzzy logic toolbox that provides MATLAB functions, apps, and Simulink blocks to analyze, design, and simulate systems based on fuzzy logic. We have designed and applied rules and received the output values.

2. Microsoft Excel is an essential tool for data collection and preparation. The software also has extensive features to perform various data analysis tasks. We used Microsoft Excel to prepare our dataset, extract values from MATLAB and store them in the dataset and perform some data analysis tasks such as root mean square error (RMSE) calculations and graph presentations.

This section discusses the implementation of the service negation frameworks. As detailed in the previous section, the framework comprises five steps. Moreover, in the service negotiation

framework, we have developed two significant reputation data-driven frameworks: a) service provider selection and b) a framework for the service provider to evaluate the suitability of accepting a request from the service consumer.

A dataset is one of the significant elements for this research as we need to collect the service reputation data and the company reputation data for both the service consumer and service provider. In the service-oriented environment, (Hussain, 2006) defines the "*reputation"* of an entity or unit as a collection of the commendations from all of the third-party recommendation agents(s) in response to the reputation querying agent's reputation query considering various factors such as dependability, consistency, reliability that produce the trustworthiness value of the reputation queried entity in a given context and in a given time slot.

In the service negotiation framework, we viewed the SDN in the SOA and delivered the services to the consumer upon service request. Therefore, we developed a complex service provider selection framework and evaluated the suitability of the service request framework to deliver the services. These frameworks are considered the service reputation data to validate the frameworks we detailed in the previous section. Section 4.1 comprehensively discusses the dataset collection and preparation, as several steps are involved in reputation data preparation.

5.5.1 Dataset Collection and Preparation

A dataset is one of the significant elements of this research as we need to collect the service reputation data and the company reputation data for both the service consumer and service provider. Service reputation and company reputation data consist of sensitive and confidential information businesses do not like to share. Therefore, collecting the service reputation and company reputation dataset is a challenge. Initially, we attempted to access and collect the company reputation and service reputation scores from the TrustRadious site (Reputation), a reputation rating site for various types of businesses. However, we need to consider the various dimensions of requirements or reputation parameters for our framework during the data collection and preparation process for our validation. The dimensions of the requirements are as follows;

- We need a reputation data set from the SDN service provider and service consumer both
- We need a generous amount of reputation data for each type.
- We need the reputation data or feedback from subsequent consumers and other organizations who have worked with the company before and provided valuable and honest feedback.
- We need to collect the reputation rating dataset on a quantitative scale; a feedback-based review cannot be processed as this is out of our scope.

Collecting and preparing datasets with the aforementioned requirements is a significant challenge that leads us to develop our in-house reputation rating dataset, also named a synthetic dataset. We primarily used a synthetic dataset. The SDN service provider reputation dataset is challenging to collect. Therefore, (Khan & Hussain, 2020) mentioned that synthetic data is essential and valuable when privacy limits the availability or usage of the data or when the data needed for a test environment does not exist.

5.5.1.1 Data Collection and Preparation for Service Provider

To implement our framework successfully, we require a dataset with various features for this approach, such as a) the count of successful transactions, b) the total number of transactions so far, c) the company's establishment year, d) reputation rating e) other companies' feedback or reputation rating on this company. Therefore, we generated our customized dataset, and a section of the data set is provided below.

The section of the dataset provided here is the service provider dataset. According to our reputation dataset requirements mentioned above, the activities that we anticipated can be categorized as a) dataset development and b) dataset preparation. The detailed steps of these activities are described below.

Service provider	Year	rating	Transaction (Ts)	successful transaction (Tn)	transaction trend	ed 15	Rank v15	ed 17	Rank17	ed 19	Rank19	ed 24	rank24	ed 26	Rank26
1	2017	1.2	14	3	21.42	6.40	174	5.09	92	11.04	208	3.16	51	13.15	214
2	2017	2.5	15	13	86.66	5.83	128	6.08	122	12.04	225	3	41	14.14	232
3	2017	3.9	28	5	17.85	11.18	358	19.02	461	25.01	464	13.34	447	27.07	465
4	2017	4	25	23	92	8.60	253	16.03	399	22.02	411	10.44	342	24.08	413
5	2010	2.3	15	4	26.66	3.60	55	10	275	13.41	253	4	80	14.86	240
6	2014	0.5	11	5	45.45	7.28	197	4.47	62	8.24	159	4	80	10.04	161
7	2012	3.1	11	6	54.54	7	179	6.32	136	8.94	165	4.47	97	10.44	164
8	2010	4.7	2	1	50	16.12	469	10.63	290	6.08	109	13.60	452	5.09	70
9	2011	4.2	4	2	50	14.03	431	8.60	237	5.09	90	11.40	373	5	54
10	2017	3.1	29	23	79.31	12.08	385	20.02	481	26.01	486	14.31	478	28.07	490
11	2013	3.6	14	2	14.28	4.12	76	7.07	155	11.40	215	1.41	9	13.15	214
12	2015	4.4	20	2	10	3.60	55	11.40	307	17.02	312	5.09	122	19	313
13	2016	3.7	28	7	25	10.77	342	19.10	463	25	463	13.15	437	27.01	464
14	2015	2.6	13	4	30.76	5.83	128	5	70	10.04	191	2.23	22	12	191
15	2012	0	18	0	0	0	1	10.81	299	15.52	289	3.60	66	17.26	286
16	2017	0.4	7	2	28.57	12.08	385	2.23	17	4.12	68	8.54	260	6.32	94

Table 5.15: A section of the service provider dataset

5.4.1.2.1 Dataset Development

The synthetic dataset development activities comprise three steps to develop a dataset that looks, feels, and behaves like a production dataset. The process follows the following steps.

Step 1: We determine and articulate the features we need to validate the framework. We tentatively design and plan to achieve the target dataset considering the framework we developed and discussed in section 3. Therefore, the following features are selected to develop the service provider dataset.

- *Service provider number:* The provider identification number that we need to identify each service provider

- *Establishment Year*: The year when the specific provider has established their company and started their business

- *Reputation Rating:* The reputation rating of the individual provider. We assume that the reputation rating is the feedback from the immediate customer of a certain organization or company.

- *Number of Transactions*: A certain service provider company performs the total number of service-related transactions where the interaction involves cost.

- *The number of successful transactions:* The total number of successful service-related transactions performed by a certain service provider where the transaction involves cost.

Step 2: In this stage, we add the values of the predefined features mentioned in step 1. Firstly, we added the first 20 providers' values.

- We used sequential numbering to identify the provider for the service provider identification number.

- We used a random year for the service provider establishment year for *establishment year* feature.

- A reputation rating scale is proposed in section 3.3.1 to receive feedback from the consumer. Based on the rating scale, we used random numbers to add the values that

represent quantifying feedback from the other organization with a cost-oriented interaction.

- The number of transactions is the feature that we assumed and interpreted as each service provider has been involved in any cost-based business interactions since they established their organization. Therefore, this is the value of the total number of transactions they have been involved in since their establishment. We chose random values for this feature.

- The number of successful transactions is another important feature that we assumed, since a certain service provider was involved in a number of business transactions. Therefore, a number of business transactions were successful among them. We define a successful business transaction as the "successful completion of a contract." This means a) the contract has not been violated or breached, b) the contract has not been broken, and c) there has been no pause in the transaction due to unexpected circumstances. We also choose random values for this feature.

Step 3: We added the values for all the features mentioned above for the first 20 service providers. The current dataset we developed for service provider reputation data is listed for 20 service providers. To better orient the framework and to better understand the developed dataset, we decided to grow the dataset, not limiting it to 20 providers. The larger dataset assists us in validating the framework and determining the accuracy of the synthetic dataset that can perform as a production dataset. Therefore, we grow the dataset that consists of 500 service provider reputation rating data.

5.4.1.2.2 Dataset Preparation

Dataset preparation is the process of preparing the dataset according to the need. We prepared the dataset using some equations mentioned in the description of the framework section (Section 3). In addition, we introduced various data analysis approaches to prepare and

experiment with the dataset and obtain the results. Thus, the dataset preparation consists of the following steps:

Step 1: Determine the transaction trend of the service provider

Firstly, we need to learn each provider's transaction trend, as the transaction trend is one of the critical parameters for our proposed service provider selection framework. The transaction trend of the service provider is depicted using equation 7. By following the transaction trend equation (equation 7), we listed the transaction trend for all the listed service providers.

Step 2: Determine the Euclidean distance of a certain service provider

In this step, we want to discover the nearest neighbour of a certain service provider by considering various parameters. Euclidean distance is a well-known and widely used method for the k-nearest neighbour (KNN) approach. We used the aforementioned nearest neighbour (NN) equation (equation 8) to discover the nearest neighbour (NN) using two parameters (establishment year and the number of transactions). As we assume that all providers listed in the dataset are active members of the service provider market, they have performed any number of business transactions since they were established. It is possible that they may have no reputation ranking or that they have not successfully finalized a successful transaction; however, they must have performed a number of business transactions.

The NN equation discovers the NN values of a certain neighbour, mainly comparing the distance from the target provider to all of the listed providers. We performed the equation for some randomly chosen providers rather than doing all of them to keep the dataset simpler and more understandable. The chosen service providers for the NN equation find and compare their distance to all 499 listed providers. Our experiment discovered the Euclidean distance for provider numbers 15, 17, 19, 24, and 24. The reason behind choosing these providers is that their rating is 0, and we want to find the nearest neighbours of these newly established providers.

Step 3: Determine the nearest neighbour of a certain service provider.

In step 2, we performed the NN equation and found the relative distance of certain providers. In this stage, to find the nearest neighbours with minimal effort, we implemented a formula that enables us to represent the NN equations' ranking. For example, if we want to see the nearest neighbour for service provider 15, the ranking '1' shows the closest neighbour for provider 15 out of 499 providers. Therefore, ranking '2' is the second closest, ranking '3' is the 3rd and ranking '4' is the 4th, and so on.

5.5.1.2 Data Collection and Preparation for Service Consumer

Service consumer data collection and preparation is a similar process that we performed to develop the service provider dataset. There are some differences since the dataset which is developed is suitable to implement and validate the framework proposed in section 3.3. The development of the dataset consists of some features and involves some co-related steps, and the steps are described in the following.

Step 1: Service consumer identification number

We followed the same process of random identification numbers used for service provider identification. Using these identification numbers, we can define a certain consumer and collect the corresponding values of the consumer. We created 500 service consumer identification numbers to create 500 pieces of service consumer information. As mentioned in the previous discussion, a more extensive dataset helps us validate the framework and increases the chances of proving the accuracy of the dataset and the framework.

Step 2: Company Reputation Rating

To define the company's reputation rating, we followed the same reputation definition, process, and rating scale described in section 3.3.1. We used random numbers to add the reputation value within the rating scale.

Step 3: Maximum Reputation

Based on our aforementioned reputation rating scale, we know the maximum reputation ranking is always 5.

Step 4 Other providers' reputation ratings.

We assumed that all the consumers have worked with various service providers in relation to receiving their services. Therefore, the consumers had business-related interactions with these providers. During their interaction, the service providers delivering their services to certain consumers have a detailed understanding of the behavioural and strategical approaches in business-related interactions. Thus, these service providers are positioning themselves to be in a reasonable position to provide a recommendation or rating for the reputation of a certain consumer organization. This reputation rating scale follows the same as that mentioned in section 3.3.1.

Step 5 Other provider's maximum reputation rating

Other providers' maximum reputation rating follows the same rating scale mentioned in section 5.3.3.1; therefore, the maximum reputation rating is added as '5'. A section of the service consumer dataset is shown in table 5.16.

sdn_consumer	company_rating	max_rating	other_provider_rating_opm	max_opr
1	2.3	5	2.7	5
2	1.1	5	1	5
3	4.2	5	5	5
4	4.8	5	5	5
5	4.9	5	5	5
6	3.1	5	3.3	5
7	3.4	5	3.5	5
8	4.5	5	5	5
9	2.8	5	3.4	5
10	3	5	2.7	5
11	2.5	5	3.6	5
12	2.5	5	3.5	5
13	2.4	5	2.3	5
14	4.1	5	2.9	5
15	2.8	5	2.8	5

16	1.8	5	2.5	5
17	2.4	5	1.8	5
18	3.1	5	3.8	5
19	1.5	5	1.6	5

Table 5.16: A section of the Service Consumer dataset

5.5.2 Experiments and Validation

Experimenting with the framework is one of the critical challenges for any research. The previous section comprehensively describes the data collection and preparation while working with synthetic datasets. As there are two critical frameworks that we developed in the service negotiation framework for personalised QoS delivery in SDN, we have categorized the experiments into two segments. The details of the experiment process are described in the following sections.

5.5.2.1 The Experiment of the Service Provider Evaluating the Suitability of Accepting a Request from the Service Consumer Framework

To experiment with the '*service provider evaluating the suitability of accepting a request from the service consumer* 'framework or, in other words, the *service request evaluation* framework, we developed a reputation rating of the service requester or service consumer (company or organization) dataset in section 5.4.3.1. The framework details in section 5.3.3 show that the framework is developed depending on three factors. The factors considered in the framework are a) reliability of the service requester, b) duration of the requester services, and c) risk exposure of the service provider in accepting the service with the required specification. We used the FIS model to experiment with the framework. In this section, we describe the experiment of the aforementioned three factors.

- To implement the FIS model, we configured three input variables for consumer company reliability, service duration and risk exposure

- We have configured the output variable according to the scale that is defined in Table 5, and the configuration design is shown in Figure 8.

- We identified the consumer company reliability value using equation 4

At this stage, we develop a reputation data-driven intelligent system called the suitability calculation that can determine the suitability of the provider and help determine whether to agree to deliver the service request. To implement this intelligent system, we followed two steps.

Step 1: We used a prevalent Mamdani method to calculate the provider's suitability and assist in critical, intelligent decision-making. The implementation results are provided later in this paper. The basic equation that we used to calculate the suitability of the provider is as follows:

$$Suitability = Reliability \ \&| \ Duration \ \&| \ Risk \ Propensity \qquad \textbf{Equation 6}$$

Step 2: We developed a fuzzy association rule and represented them in a multidimensional form. In other words, this is a rule matrix of the fuzzy interference systems (FIS) to map the input variables and receive the output values that assist in intelligent decision-making.

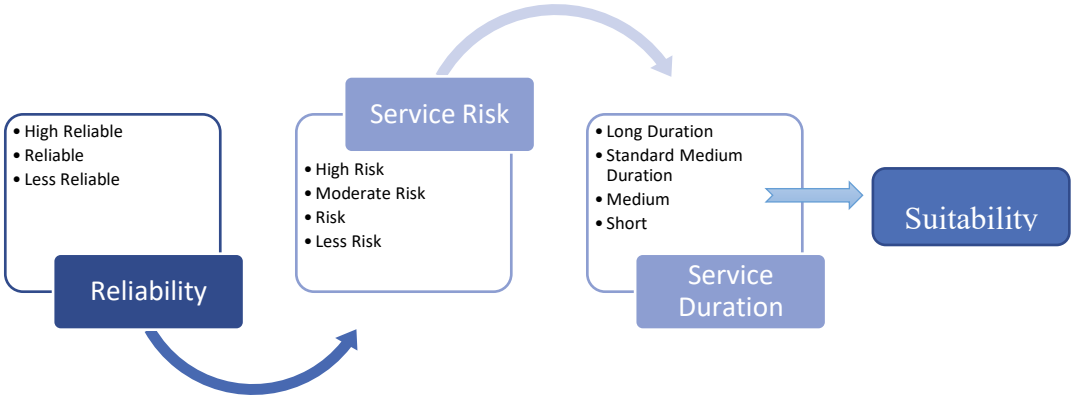


Figure 5.12: Suitability Calculation Framework

Step 3: We perform the permutation and construct the rules in the FIS) We mapped the input variable strength and the output generated according to that strength. We discuss in detail the input variable strength in the following.

Three distinct factors are depicted in Figure 5.12. Of these, we choose company reliability which has the highest strength. According to our assessment, company reliability is the first factor that every provider company should consider before considering any other factors. Thus, we illustrated the reliability factor as (Figure 5.12) the first and most profound step and represented it in the deepest colour code. Let us identify how the output influences input strength. Let us consider a consumer company named "A" looking for a service for a **concise duration.** The type of service that the consumer company is looking for is costly. To deliver the services that company A requires, provider company B needs to spend most of its resources and put their company in a **high-risk** Zone. However, provider B identified that Consumer A is a very reliable company using its historical profile information. Therefore, the fuzzy association rules determine the result as company A is suitable for provider B to deliver services. According to this process, service risk carries the second-highest strength in this framework. The results of the framework are described in the results section.

1. If (Service_Duration is TooShort) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is LessSuitable) (1)
2. If (Service_Duration is TooShort) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is ModerateSuitable) (1)
3. If (Service_Duration is TooShort) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
4. If (Service_Duration is TooShort) and (Service_Risk is Risk) and (Consumer_Company_Reliability is LessReliable) then (output1 is LessSuitable) (1)
5. If (Service_Duration is TooShort) and (Service_Risk is Risk) and (Consumer_Company_Reliability is Reliable) then (output1 is ModerateSuitable) (1)
6. If (Service_Duration is TooShort) and (Service_Risk is Risk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
7. If (Service_Duration is TooShort) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is LessSuitable) (1)
8. If (Service_Duration is TooShort) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is Suitable) (1)
9. If (Service_Duration is TooShort) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
10. If (Service_Duration is TooShort) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is ModerateSuitable) (1)
11. If (Service_Duration is TooShort) and (Service_Risk is Risk) and (Consumer_Company_Reliability is Reliable) then (output1 is ModerateSuitable) (1)
12. If (Service_Duration is TooShort) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is ModerateSuitable) (1)
13. If (Service_Duration is Short) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is LessSuitable) (1)
14. If (Service_Duration is Short) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is ModerateSuitable) (1)
15. If (Service_Duration is Short) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
16. If (Service_Duration is Short) and (Service_Risk is Risk) and (Consumer_Company_Reliability is LessReliable) then (output1 is LessSuitable) (1)
17. If (Service_Duration is Short) and (Service_Risk is Risk) and (Consumer_Company_Reliability is Reliable) then (output1 is ModerateSuitable) (1)

11. If (Service_Duration is TooShort) and (Service_Risk is Risk) and (Consumer_Company_Reliability is Reliable) then (output1 is ModerateSuitable) (1)
12. If (Service_Duration is TooShort) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is ModerateSuitable) (1)
13. If (Service_Duration is Short) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is LessSuitable) (1)
14. If (Service_Duration is Short) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is ModerateSuitable) (1)
15. If (Service_Duration is Short) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
16. If (Service_Duration is Short) and (Service_Risk is Risk) and (Consumer_Company_Reliability is LessReliable) then (output1 is LessSuitable) (1)
17. If (Service_Duration is Short) and (Service_Risk is Risk) and (Consumer_Company_Reliability is Reliable) then (output1 is ModerateSuitable) (1)
18. If (Service_Duration is Short) and (Service_Risk is Risk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
19. If (Service_Duration is Short) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is ModerateSuitable) (1)
20. If (Service_Duration is Short) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is Suitable) (1)
21. If (Service_Duration is Short) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
22. If (Service_Duration is Short) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is ModerateSuitable) (1)
23. If (Service_Duration is Short) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is Suitable) (1)
24. If (Service_Duration is Short) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
25. If (Service_Duration is Medium) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is LessSuitable) (1)
26. If (Service_Duration is Medium) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is ModerateSuitable) (1)
27. If (Service_Duration is Medium) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)

28. If (Service_Duration is Medium) and (Service_Risk is Risk) and (Consumer_Company_Reliability is LessReliable) then (output1 is LessSuitable) (1)
29. If (Service_Duration is Medium) and (Service_Risk is Risk) and (Consumer_Company_Reliability is Reliable) then (output1 is ModerateSuitable) (1)
30. If (Service_Duration is Medium) and (Service_Risk is Risk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
31. If (Service_Duration is Medium) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is ModerateSuitable) (1)
32. If (Service_Duration is Medium) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is Suitable) (1)
33. If (Service_Duration is Medium) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
34. If (Service_Duration is Medium) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is ModerateSuitable) (1)
35. If (Service_Duration is Medium) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is Suitable) (1)
36. If (Service_Duration is Medium) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is High) (1)
37. If (Service_Duration is StandardMedium) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is LessSuitable) (1)
38. If (Service_Duration is StandardMedium) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is ModerateSuitable) (1)
39. If (Service_Duration is StandardMedium) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
40. If (Service_Duration is StandardMedium) and (Service_Risk is Risk) and (Consumer_Company_Reliability is LessReliable) then (output1 is LessSuitable) (1)
41. If (Service_Duration is StandardMedium) and (Service_Risk is Risk) and (Consumer_Company_Reliability is Reliable) then (output1 is ModerateSuitable) (1)
42. If (Service_Duration is StandardMedium) and (Service_Risk is Risk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
43. If (Service_Duration is StandardMedium) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is ModerateSuitable) (1)
44. If (Service_Duration is StandardMedium) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is Suitable) (1)

44. If (Service_Duration is StandardMedium) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is Suitable) (1)
45. If (Service_Duration is StandardMedium) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
46. If (Service_Duration is StandardMedium) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is ModerateSuitable) (1)
47. If (Service_Duration is StandardMedium) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is Suitable) (1)
48. If (Service_Duration is StandardMedium) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is High) (1)
49. If (Service_Duration is Long) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is LessSuitable) (1)
50. If (Service_Duration is Long) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is ModerateSuitable) (1)
51. If (Service_Duration is Long) and (Service_Risk is HighRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
52. If (Service_Duration is Long) and (Service_Risk is Risk) and (Consumer_Company_Reliability is LessReliable) then (output1 is ModerateSuitable) (1)
53. If (Service_Duration is Long) and (Service_Risk is Risk) and (Consumer_Company_Reliability is Reliable) then (output1 is Suitable) (1)
54. If (Service_Duration is Long) and (Service_Risk is Risk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is Suitable) (1)
55. If (Service_Duration is Long) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is ModerateSuitable) (1)
56. If (Service_Duration is Long) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is Suitable) (1)
57. If (Service_Duration is Long) and (Service_Risk is MediumRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is High) (1)
58. If (Service_Duration is Long) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is LessReliable) then (output1 is ModerateSuitable) (1)
59. If (Service_Duration is Long) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is Reliable) then (output1 is Suitable) (1)
60. If (Service_Duration is Long) and (Service_Risk is LowRisk) and (Consumer_Company_Reliability is Highly_Reliable) then (output1 is High) (1)

Figure 5.13: 60 rules to operate the FIS model to conduct experiments on the framework.

5.5.2.2 The Experiment of the Reputation-based Provider Selection Framework

To experiment with the reputation-based provider selection framework, we developed a reputation rating of the service provider dataset in section 4.1. During the dataset development, we conducted all of the necessary tasks for preparing the dataset to experiment with the framework. From the discussion of the framework (section 3.4), we segmentized it into two categories of service providers according to their existence in the current market; a) established SDN providers are those who have been in the market for more than 2 years, and b) emerging providers who are relatively new to the market and do not have any previous transaction history.

Step 1: Determine a suitable threshold value

We implemented the framework using the FIS model and designed the input and output variables to receive the output. Section 5.3.4.1 uses the FIS to design the input variables according to the predefined scale (Table 5.6). We designed the output variable where we have defined two threshold values such as '45' and '35' using a reasonable randomly chosen number.

At this point, we need to determine the most suitable threshold number between two numbers that have a lower error rate. To compare the errors between these two numbers, we first measure the errors by performing the following process.

- In the first stage, we segmented the dataset and marked the segment to create the time series dataset. In order to do this, we chose 10 records for a one-time series dataset; for example, record numbers 2 to 11 are denoted as time series 1 or T1, and record numbers 12 to 21 are denoted as time series 2 or T2, and the segmentation process continues. We created 6 time-series segments to perform the next stage of the experiment.

- As this is a synthetic dataset, we have no true value against which we can compare our predictive value with the true values. Therefore, we generated our 'trend_cohort_true_value,' the average transaction trend value for each time series.

- We used FIS to determine the output value when the threshold is 45 and recorded these values for every provider from all of the time series (T1 to T6).

- Similarly, we used FIS to determine the output value when the threshold is 35 and recorded these values for every provider from all of the time series (T1 to T6).

- In the next stage, we used the RMSE to predict the errors by employing the residuals' standard deviation. We calculated the RMSE between the trend_cohort_true_value and the fuzzy model output when the threshold is 45.

- Similarly, we calculated the same RMSE between the trend_cohort_true_value and the fuzzy model output when the threshold is 35.

- The results show that for threshold 35, the RMSE is 7.19, and for threshold 45, the RMSE is 8.82. Therefore, we summarize that threshold 35 is suitable for generating fewer errors.

The model is trained using the time-series value and the expectation that the line would progress higher as the model used to learn from the training dataset, and then gradually, the line should be downgraded to lower as the model test the last portion of the time-series dataset.

Step 2: Fuzzy Logic-based experiment for emerging providers.

We employed FIS to experiment with our framework in selecting a suitable provider from a list of emerging providers. The definition of an emerging provider is discussed in section 3. To implement the framework, we designed two input variables named the “transaction_trend” and “reputation rating”, and one output variable named “Accept/Reject decision”. The design is illustrated in section 3. We configure the rules and define the threshold. According to the rules, the system generates the output. We recorded the output in terms of thresholds 45 and 35 to perform the experiment detailed in steps 1 and 4.

Step 3: Fuzzy Logic-based experiment for new providers.

As they have no reputation rating (one input variable), the FIS cannot perform the calculation in terms of new service providers. Therefore, we perform the implementation by referring to the four sub-steps described in section 5.3.4.2, and the implementation steps are as follows:

Similar neighbours' discovery:

We performed the Euclidean distance calculation to determine the similarity of the new target provider to all the other providers. We apply NN formula 8 to find similar neighbours in the spreadsheet and generate the NN values.

After receiving the NN values, we ranked the values and found the acceptable closest service provider of the new target provider.

Define the best K value:

Using Python, we find the best k-values for our developed dataset. The best K-value that we found for this experiment is 10, which means to perform the next task, we consider the top nearest 10 provider values of a certain new provider.

Transaction trend of similar neighbours:

In this stage, we determine the transaction trend of a certain new service provider. As our best k-value is 10, we used the transaction trend of the top nearest 10 service providers using formula 10 described in section 5.3.4.2 to find the transaction trend of a certain new service provider. Therefore, this approach is used to realize the transaction trend of a provider who has limited information publicly available about their reputation and success rate.

The decision to approve or reject the new provider:

After identifying the transaction trend of a service provider who is relatively new to the market, the consumer needs to make a decision as to whether they should choose a certain new service provider with which to build a trust relationship and to deliver their requested services. To make this decision, we used formula 11 and implemented the formula in a FIS. As before, we used the Mamdani FIS, where we created the input variable for "threshold". Another input

variable is the transaction trend of the provider. The trend value can be identified in the previous step.

We generated the output variable as accept and reject, and the scale of the output variable is detailed in Table 5.13. The FIS design is shown in Figure 5.9. We collected the results of the output values in a spreadsheet.

Step 4: Validate the accuracy of the framework

We mentioned the rationale for using our dataset (synthetic dataset); thus, we have limited opportunities to validate our dataset and the framework with a real-life production dataset and compare the framework with true values. In these circumstances, we proposed a time series-oriented dataset to validate the framework that can be performed using a portion of the time-series dataset to train the framework model and a portion of the dataset is used for testing purposes and giving the outcome. The proposed validation task can be performed using the following actions.

- We created our dataset from a set of 20 service provider datasets, then generated the 500 service provider dataset out of the first 20 using an arbitrary approach.

- We generated time-series data from the dataset, and the details are discussed in the above step.

- In the validation spreadsheet, the time-series dataset represents the list of the service providers' identification numbers with their associated information

- The actual value is represented as the ‘trend_cohort_true_value,’ and the process of determining the ‘trend_cohort_true_value’ is described above in step 1 and section 5.2.2.2.

- We applied FIS to produce the output of the framework that, considering the threshold value is 45

- Similarly, we employed FIS to produce the output of the framework considering the threshold value is 45

- At this point, we need to determine the errors between the trend_cohort_True_value and the FIS-produced outcome when the threshold is 45 and 35, aligned with the time series. This means we employed the RMSE for each time series and labelled them according to the time series name.

- Therefore, we receive RMSE for T1 to T6 in terms of threshold (45 and 35) values.

In the next stage, we separated the time series values for threshold 45 and threshold 35. We developed a line graph to understand the orientation of the time series.

5.5.3 Results and Discussions

The framework experiment and validation are described in the previous section (section 4.2). In this section, we discuss the results, followed by discussing the framework's success in delivering the services that enhance the trust relationship between the service provider and consumer by employing personalised service delivery of the service-oriented SDN. We explain the results of the two essential sub-frameworks in two sub-sections below, where the framework is explained in section 5.3. Firstly, we describe the results of the service provider evaluating the suitability of accepting a request from the service consumer framework, in other words, the service request evaluation, and secondly, we describe the results of the service provider selection framework.

5.5.3.1 The Results of the Service Provider Evaluating the Suitability of Accepting a Request from the Service Consumer Framework

During the *service request evaluation framework* experiments, the rules we configured in the FIS are co-related and developing outcomes.

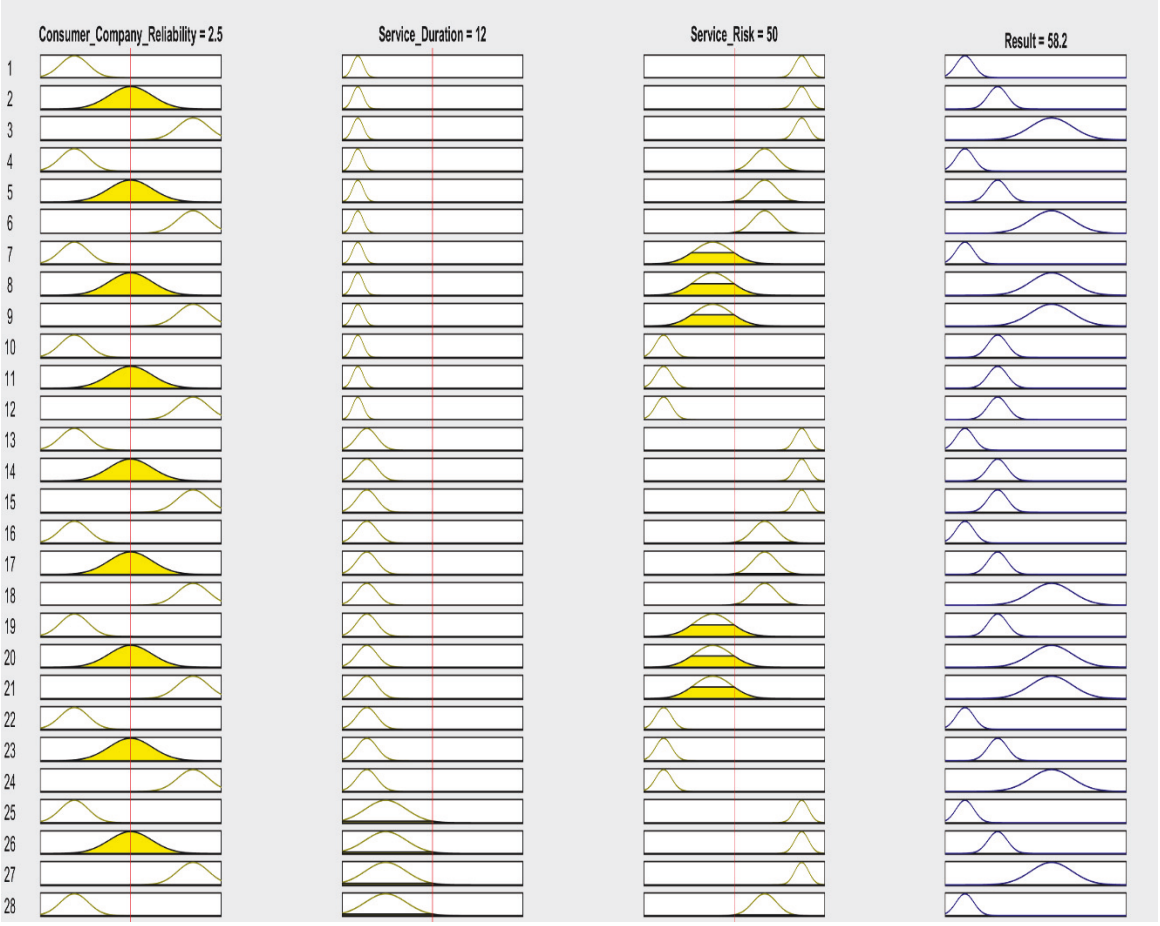


Figure 5.14: The rule co-relation of the FIS to determine the suitability of the service request.

The rules co-ordination and formulation of the results are illustrated in figure 5.14 above, which shows that if the service is requested for 18 months, the service risk is measured as 38, and the consumer company reliability is 4.25; then the output generates a value of 0.864, which shows that the certain service request is evaluated as reasonable for a provider who is evaluating the request.

The three-dimensional (3D) surface view is one of the valuable features to demonstrate the coordination of the input variables in FIS and formulate the results. We captured some 3D surface views of the results to prove our proposed framework's proof of concept. Our FIS-based

experiments consist of three input variables. The 3D surface view can demonstrate two input variables at a time; therefore, we used any two input variables and showed the dependability and co-relation of the output from any input variables.

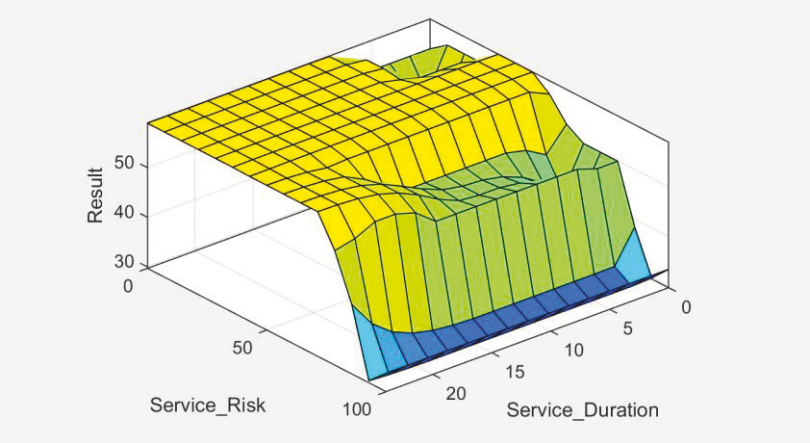


Figure 5.15: 3D surface view of the output from two input variables, service risk and service duration.

Figure 5.15 above shows a 3D surface view of the framework using two input variables (service risk and service duration). As mentioned above, the figure shows the coordination of the service risk and service duration input variables and generates the output of the consumer's suitability. The surface view also demonstrates that if the service is requested for a long duration, the risk of delivering the service is identified as relatively low. This demonstrates a comparatively higher chance of evaluating the service request as suitable. On the other hand, it also represents a relatively minor possibility of evaluating the service request as suitable if the service duration request is deficient and the service risk is moderate.

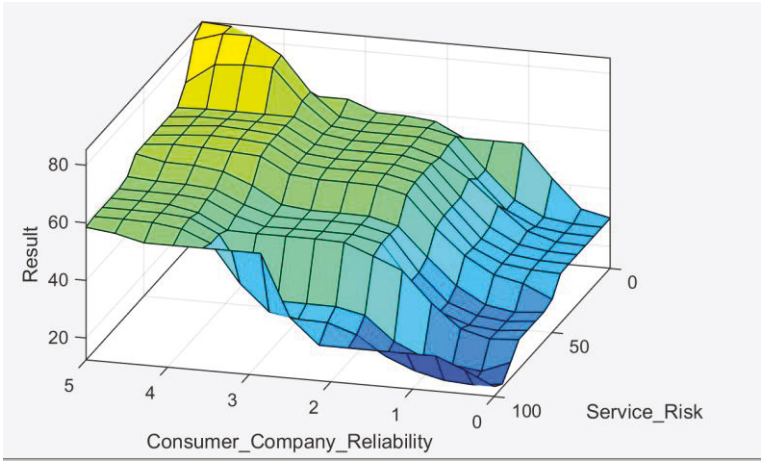


Figure 5.16: 3D surface view of the output from two input variables, service risk and consumer company reliability.

Similarly, figure 5.16 above shows a 3D surface view of the framework using two input variables (service risk and consumer company reliability). The figure shows the coordination of the service risk and the consumer company reliability as input variables and generates the output of the consumer's suitability. In addition, the surface view demonstrates that if the service is requested from a consumer company whose reputation is relatively high and to deliver the service, the risk is identified as relatively low; then, there is a comparatively higher chance that the service request will be evaluated as suitable. On the other hand, it also represents a relatively more minor possibility of evaluating the service request as suitable if the reputation of the consumer company who requested the service is inadequate and the service risk is identified as high.

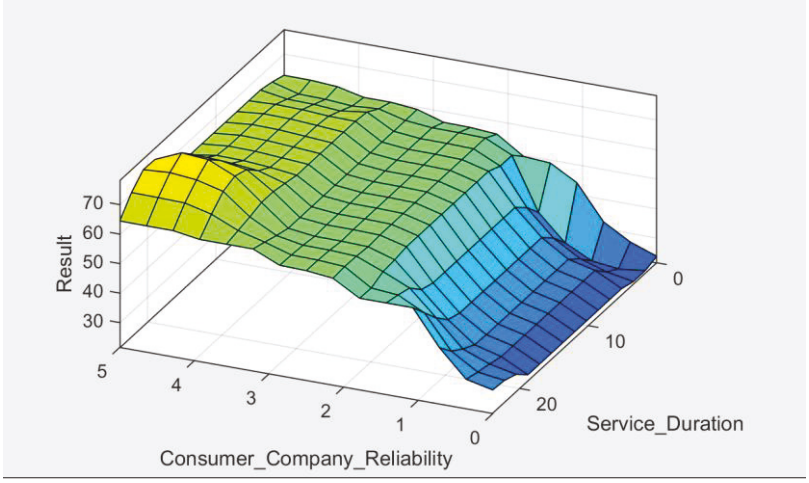


Figure 5.17: 3D surface view of the output from two input variables, consumer company reliability and service duration.

Likewise, figure 5.17 above shows a 3D surface view of the framework using two input variables (service duration and consumer company reliability). The figure shows the coordination of the service duration and the consumer company reliability as input variables and generates the output of the consumer's suitability. In addition, the surface view demonstrates that if the service is requested from a consumer company whose reputation is relatively high and the service is requested for a longer duration, there is a comparatively higher chance of evaluating the service request as suitable.

5.5.3.2 The Results of the Reputation-based Provider Selection Framework (for both existing and emerging service providers)

The reputation-based provider selection framework experiments are organized into two types as we developed two different frameworks for existing and emerging service providers. Therefore, in this section, we discuss the following results for both types of providers. In addition, we discussed the configuration of the FIS rules that relate to each other and developed outcomes similar to the previous section. Hence, we used the same approach and generated the results below, followed by a discussion.

5.5.3.2.1 Provider selection from an Existing Service Provider

The previous section describes the experiment details of the framework. During the framework experiments, the rules configured in the FIS co-relate to each other and generate the output. The coordination of the rules and the formulation of the results are demonstrated in figure 5.18 below. We designed the output membership function and scaled them. If the output value falls from 0 to 40%, the framework will recommend them to reject the provider. On the other hand, if the output value falls between 40% - and 100%, the framework would recommend them to accept the provider. Hence, figure 5.18 below shows that if the reputation rating is 2.5 and the transaction trend is 55, then the output generates a value of 70, which depicts that a certain service provider is considered a suitable service provider by a certain consumer employing the service provider selection process.

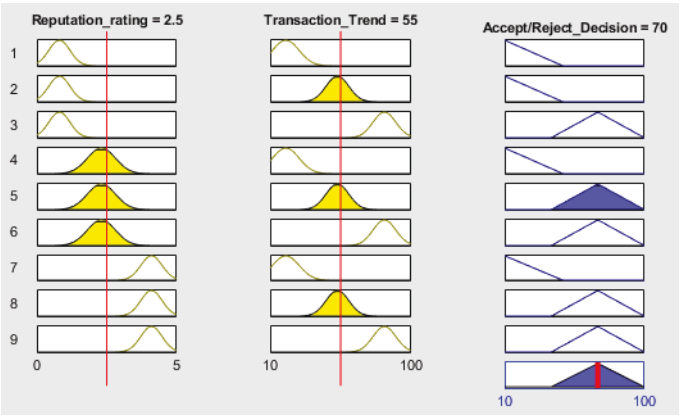


Figure 5.18: Emerging service provider selection

Figure 5.19 below shows a 3D surface view of the framework using two input variables (transaction trend and reputation rating). As mentioned in the previous section, the figure shows the coordination of the transaction trend (x-axis) and reputation rating (y-axis) input variables and generates the Accept/Reject decision-making output.

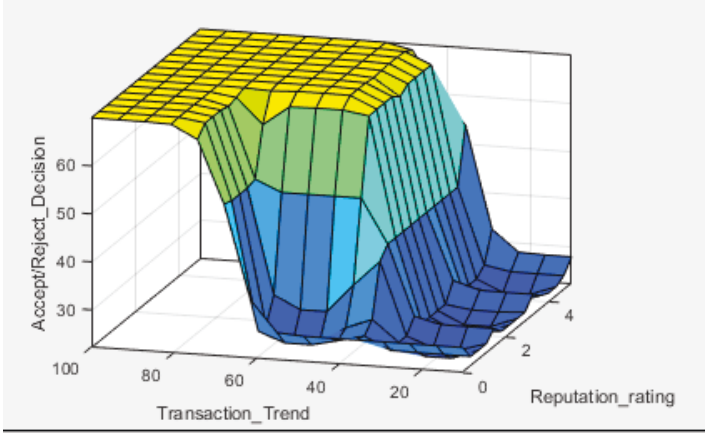


Figure 5.19: 3-dimensional surface view where the x input is Transaction Trend, y input is Reputation rating, and z output is Accept/Reject decision.

The surface view also demonstrates that if a certain service provider's transaction trend is high and the reputation ranking is high, the provider receives an acceptance result from the intelligent framework. On the other hand, it also represents a relatively more minor possibility of accepting a certain service provider as a suitable one if the transaction trend is minimal and the service reputation rating is identified as moderate or vice versa.

5.5.3.2.2 Provider section from an emerging service provider

We defined the *emerging service provider* and the framework to evaluate them in our framework detailed in section 3.4.2 and explicitly discussed the experiments and validation in the experiment section. Since our experiments consist of a few steps, we discuss the results accordingly.

Similar Neighbour Discovery:

Figure 5.20 below demonstrates the output of the similar neighbour discovery of provider number 15. We used 500 provider datasets for our experiment; however, for figure 5.20 below, we used the first 50 providers because if we include all of the providers, the result is identified

as very dense and hard to observe. The result demonstrates that, for provider number 15, the next provider is provider number 22 within the selected dataset (provider number 1 to provider number 50).

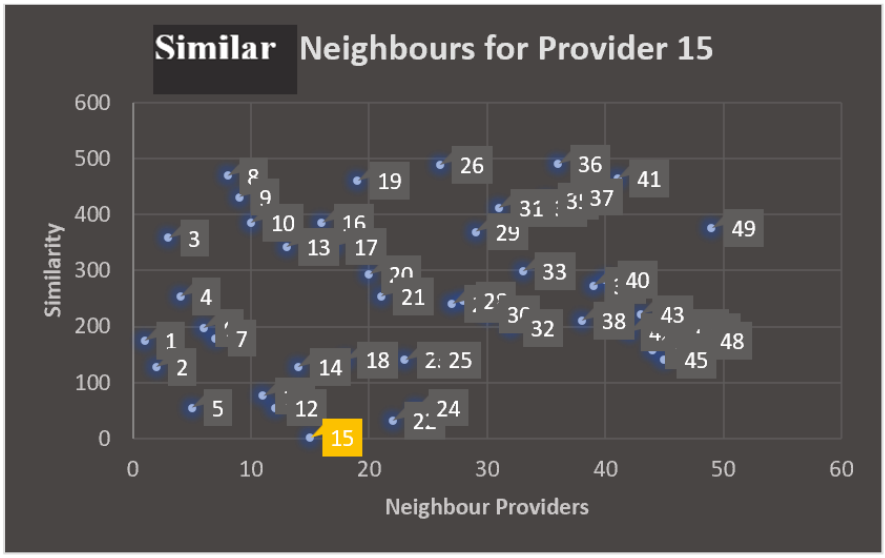


Figure 5.20: Similar neighbour discovery for provider 15

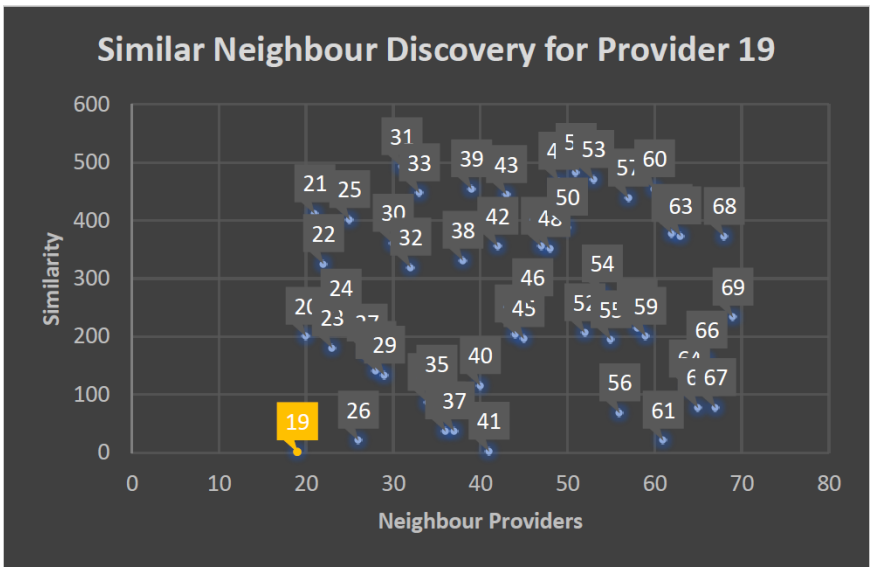


Figure 5.21: Similar Neighbour Discovery for provider 19

Similarly, the figure above 5.21 demonstrates the similar neighbour discovery for provider number 19. For this result, we used providers from provider number 19 to provider number 70. The result shows that the closest neighbour to provider 19 is provider 41 (within the sample range). Furthermore, we added a sample figure 5.22 of the result integrating all data, which is demonstrated below.

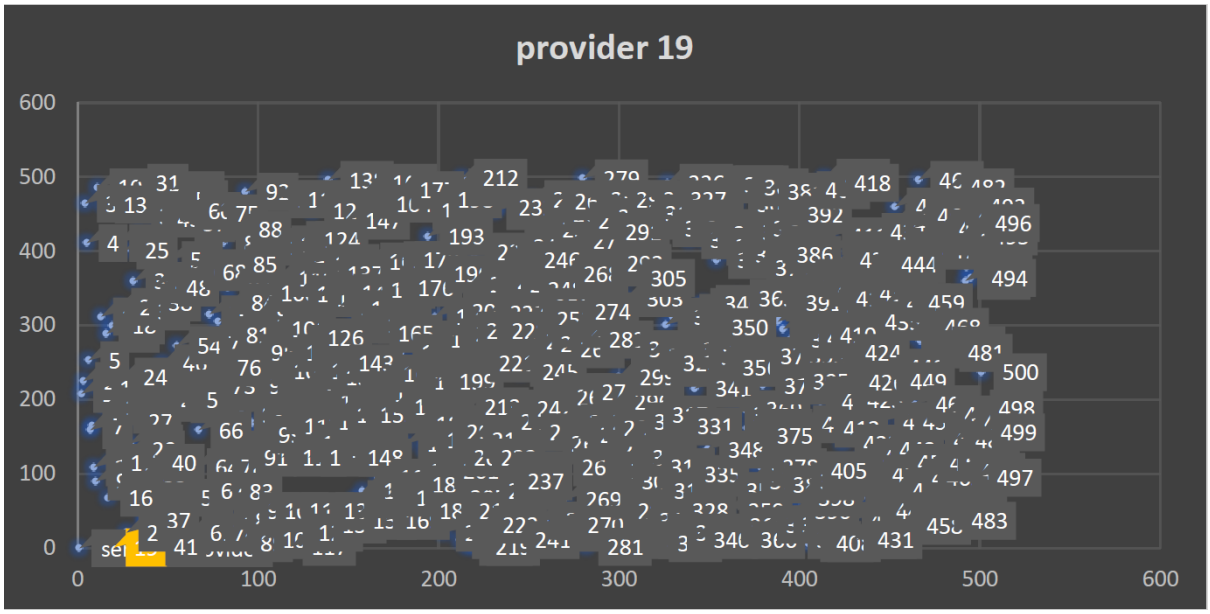


Figure 5.22: Similar neighbour for provider 19 using the whole dataset.

The decision to approve or reject the new provider

To demonstrate the result, we selected a couple of emerging service provider numbers and tested their results using the FIS model. The emerging providers that we selected are providers numbered 15, 17, 19, 24, and 27. In the experiment section, we describe the process we followed to collect the transaction trend of the emerging providers. In this section, we observe the transaction trend of the emerging providers and the coordination of the input variable transaction trend and threshold to produce the outcome.

For Provider Number 15:

The transaction trend for provider number 15 is identified as 62.11730306157551. We used two thresholds to observe the result and determine the better threshold considering the low error rate, described later in this section. Figure 5.23 below shows the co-relation of the input variable transaction trend and threshold to generate the output of a value representing an approval or rejection of a certain emerging service provider. Considering the process of our proposed model, it generates the approval decision for service provider 15 based on the threshold (table 5.7).

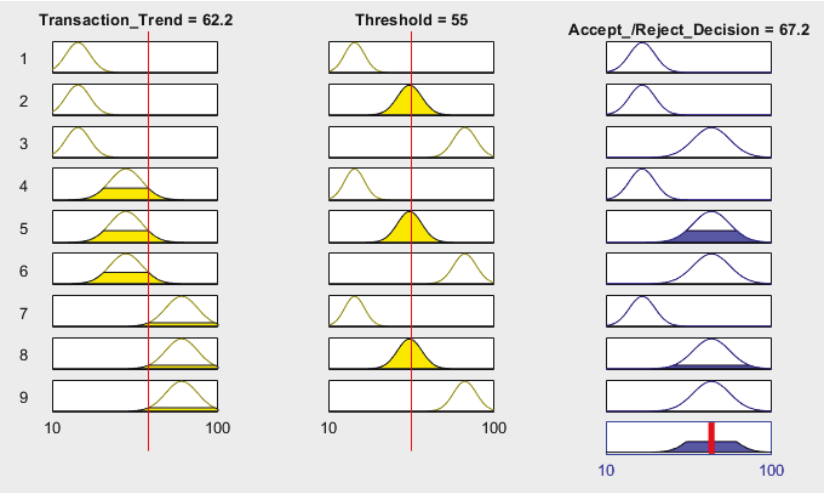


Figure 5.23: The rule co-relation of the FIS to find the outcome for provider number 15.

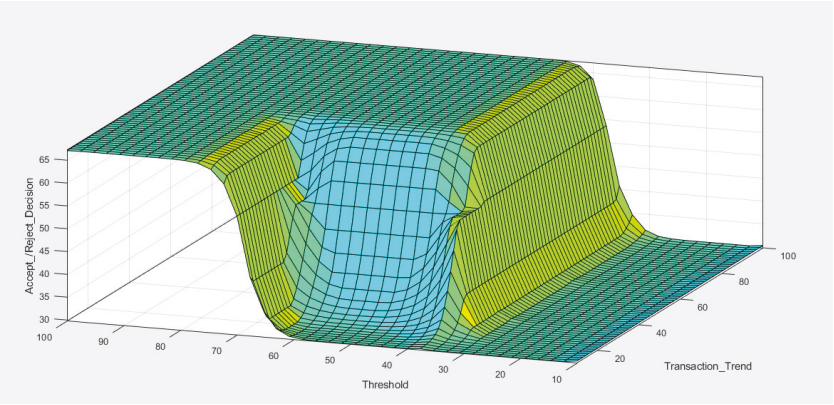


Figure 5.24: 3D surface view of the output of two input variables (transaction trend and threshold) for provider number 15

For Provider number 17

The transaction Trend for provider number 17 is identified as 47.72222222222222. Similarly, based on the threshold (table 5.7), our proposed model generates an approval decision for service provider 17. Figure 5.25 and 5.26 shows the FIS model's rules co-relation and 3D surface view. In addition, as mentioned above, the outcome will change according to the threshold which is defined for the framework.

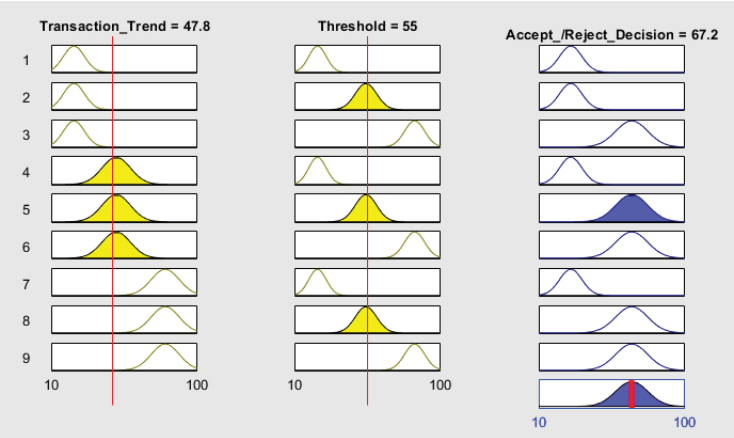


Figure 5.25: The rule co-relation of the FIS to find the outcome for provider number 17.

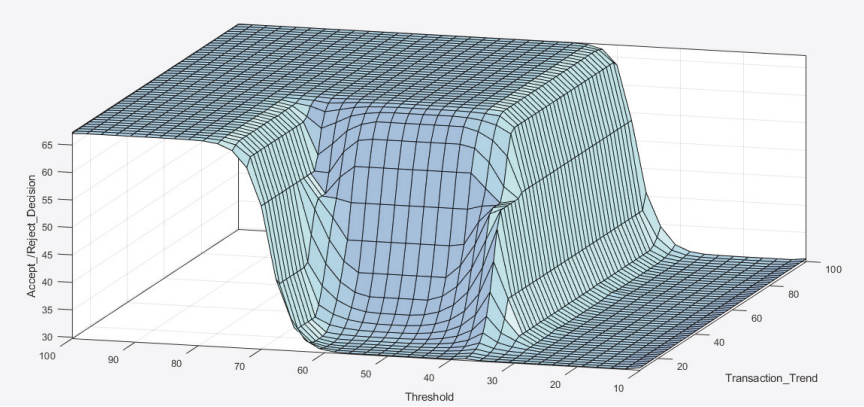


Figure 5.26: 3D surface view of the output of two input variables (transaction trend and threshold) for provider number 17

For Provider Number 19

The transaction trend for provider number 19 is identified as `20.666666666666664`. Similarly, based on the threshold (table 5.7), our proposed model generates the rejection decision for service provider 19. Figures 5.27 and 5.28 show the FIS model's rules co-relation and 3D surface view.

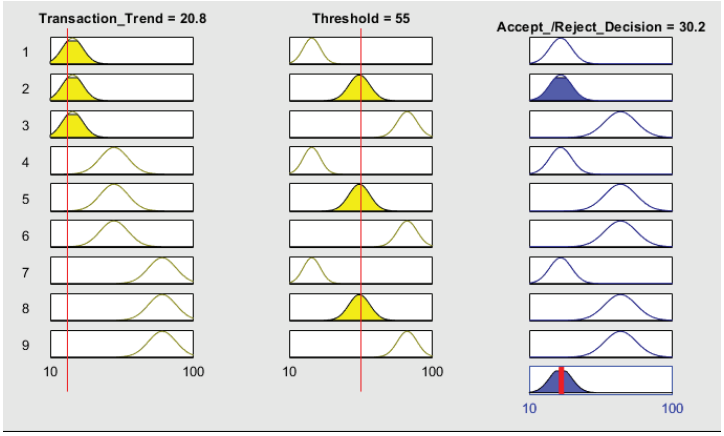


Figure 5.27: The rule co-relation of the FIS to find the outcome for provider number 19

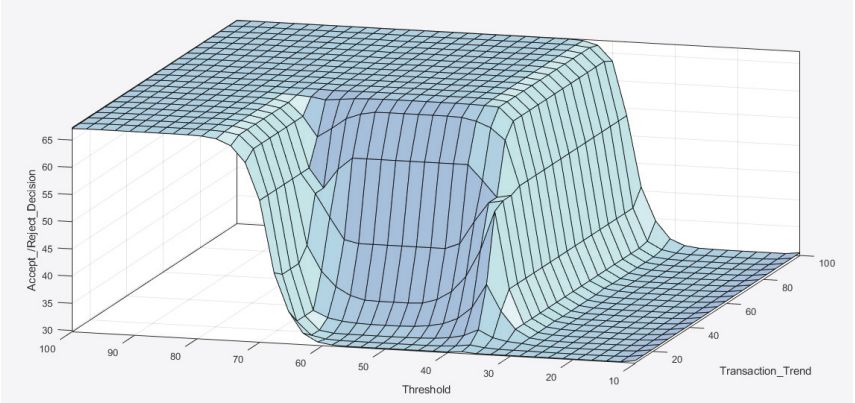


Figure 5.28: 3D surface view of the output of two input variables (transaction trend and threshold) for provider number 19

For Provider Number 24

The transaction trend for provider number 24 is identified as 34.57142857142857. Similarly, based on the threshold (table 5.7), our proposed model generates the approval decision for service providers 24. Figures 5.29 and 5.30 show the FIS model's rules co-relation and 3D surface view.

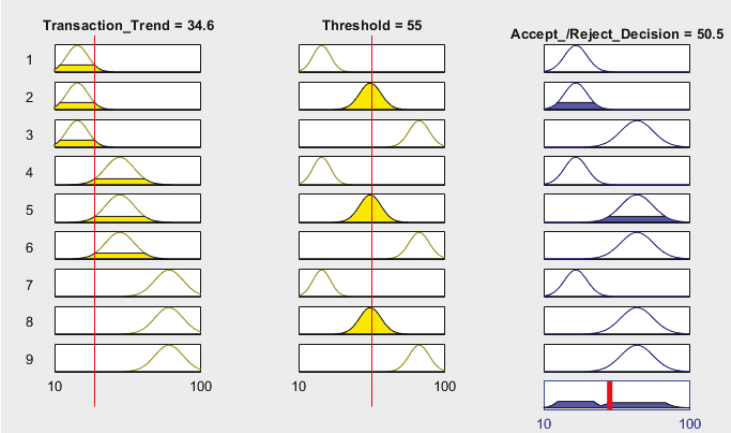


Figure 5.29: The rule co-relation of the FIS to find the outcome for a provider number.

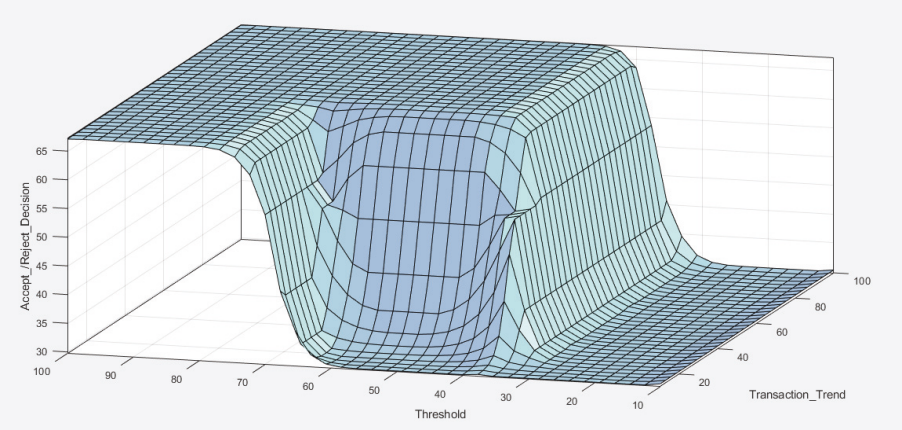


Figure 5.30: 3D surface view of the output of two input variables (transaction trend and threshold) for provider 24

For provider Number 26

The transaction trend for provider number 26 is identified as 5.0. Similarly, based on the threshold (table 5.7), our proposed model generates the rejection decision for service provider 24. Figures 5.31 and 5.32 show the FIS model's rules co-relation and 3D surface view.

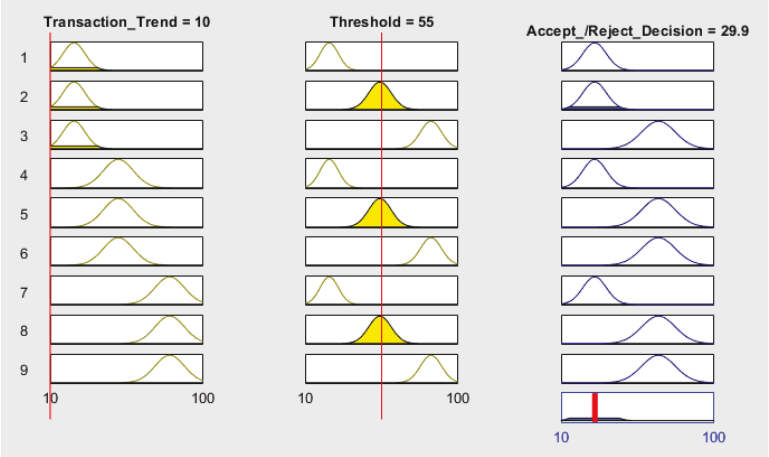


Figure 5.31: The rule co-relation of the FIS to find the outcome for provider number 26.

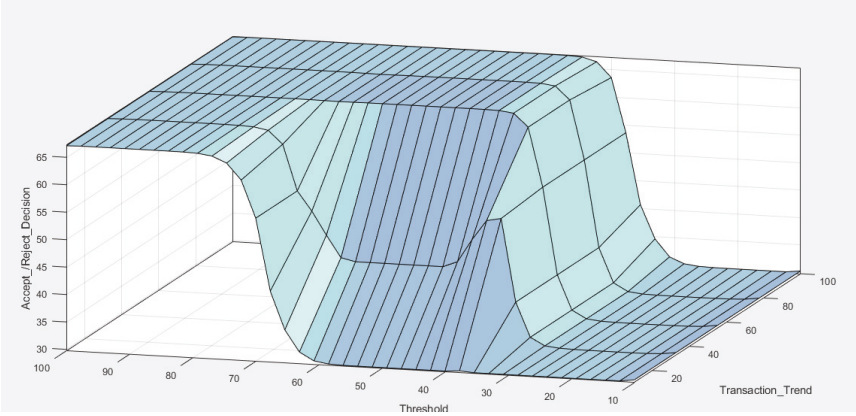


Figure 5.32: 3D surface view of the output of two input variables (transaction trend and threshold) for provider number 26

5.5.3.3 The Result to Determine a Suitable Threshold Value

The result that we used to determine a suitable threshold value for our framework is as follows. We collected the trend_cohort_True Value and equation predicted value, the FIS model output for threshold 45, and the FIS model output for threshold 35. Then, we evaluated the values using RMSE to find the lowest error. Figure 5.33 shows the values collected from our FIS model and equation (predicted value) results. Figure 5.34 shows the RMSE results, which indicate that the value of threshold 35 is able to minimize the error better compared to threshold 45. Furthermore, the result from section 4.3.4 also shows that 35 is a more suitable threshold compared to 45.

trend_cohort True value	trend_pred	Fuzzy Res/45Trs	Fuzzy Res/35Thrs
24.18247542	62.11730306	22.5	26.8
24.18247542	47.72222222	22.5	26.8
24.18247542	20.66666667	22.5	26.8
61.92929293	34.57142857	59.3	47.8
61.92929293	5	59.3	47.8
44.63701879	60.50501253	59.3	47.8
44.63701879	6.666666667	58.9	47.8
44.63701879	33.83333333	58.9	47.8
51.73635414	30.54212454	59.3	47.8

Figure 5.33: Collection of the values for the emerging service provider

rmse_trend_pred	rmse_fr45	rmse_fr35
30.28454818	8.827957835	7.190552261

Figure 5.34: RMSE results

Figure 5.33 shows the RMSE calculation results for the time-series value, and threshold 35 shows better results shown in figure 5.34.

5.5.3.4 The Result of Validating the Developed Dataset

The proposed reputation rating dataset validation is another critical task we performed in our implementation. As mentioned in the experiment section, we train our model first using our time series dataset and then test our model using the last time series of the dataset. According

to the research approach, we identified and graphically represented the time series data distribution on RMSE for threshold 45. The result is shown in figure 5.35 below.

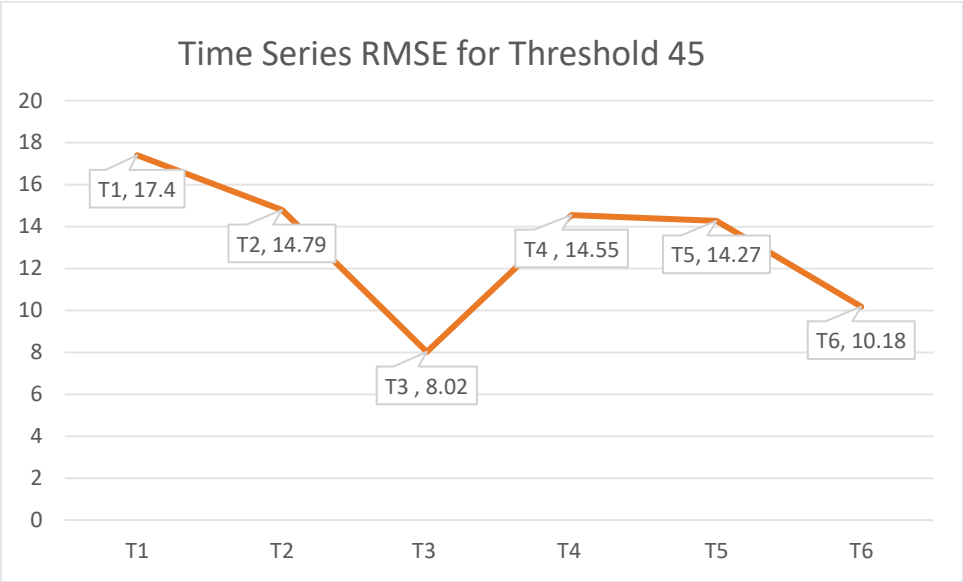


Figure 5.35: Time Series RMSE for Threshold 45

Similarly, we identified and graphically represented the time series data distribution on RMSE for threshold 35. The result is shown in figure 5.36. The distribution for threshold 35 is a more coherent result compared to threshold 45. Moreover, the results show that 35 is a more suitable threshold compared to 45.

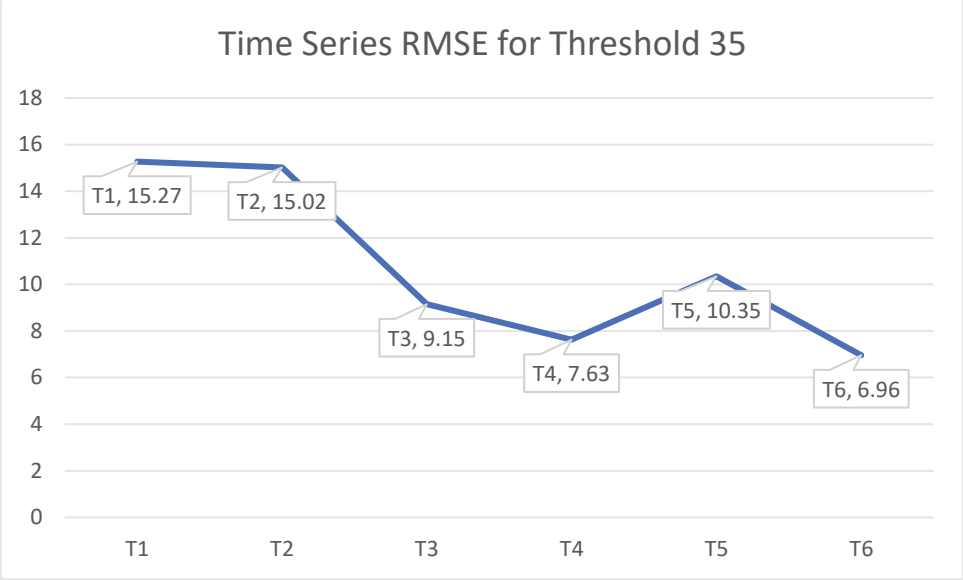


Figure 5.36: Time series RMSE for threshold 35

5.5.3.5 Discussion

A service negotiation framework for personalised service delivery in an SOA-based SDN is developed and demonstrated in this chapter. Moreover, we argued that the developed framework could significantly enhance the trust relationship among all interacting parties.

Researchers in the literature have proposed frameworks to personalise service delivery in SOA to enhance trust among interacting agents. Most of these frameworks consist of an independent agent, the broker, who liaises with the service consumer and service provider to utilise a reputation-based selection process that facilitates intelligent decision-making for both. However, for such SOA-based mechanisms to be applied in SDN to achieve personalised service delivery, the following specific requirements (defined in section 1) need to be met:

Requirement 1: Ensure that the intermediate agent or the broker has a precise and comprehensive understanding of the requested services by the consumer (hereafter considered R1).

Requirement 2: Assisting the service providers to make an informed decision as to whether to accept a service request or not (hereafter considered R2).

Requirement 3: Assisting the service users in making an optimal decision of selecting a suitable provider from a list of interested providers (hereafter considered R3).

In the domain of SOA-based SDN, existing provider selection approaches do not consider these requirements and hence may not lead to making an informed selection, both from the perspective of the service consumer and the service provider. This paper addresses this gap by proposing a framework that can provide personalised service delivery in SDN. For the service provider, the proposed approach uses a reputation-driven intelligent framework to decide whether to accept a request from a specific consumer or not. For the service consumer, the proposed approach assists in clarifying the service requirements and uses a reputation data-driven intelligent framework to select the most appropriate service provider to interact with. In doing so , it is clearly evident that the developed framework assists in enhancing the trust relationship between the agents which will be utilised in forming future service agreements.

5.6 Conclusion

Today, SLA management is a fundamental challenge for a service-oriented business model, particularly in the SDN business environment. This paper introduced a service negotiation framework with experiment results which is an initial part of the SLA. The literature review findings demonstrate that there is currently no research in managing SLA to ensure QoS in SDN [7]. Moreover, there is little research on SOA-based service management in SDN that can provide personalised service delivery. An SLA negotiation framework was introduced to address these gaps as the service negotiation occurs before the SLA is developed to avoid conflicts. The framework can achieve personalised service delivery in SDN, which is an advanced feature of SOA. Moreover, this research proposed a reputation rating based on service provider selection and service consumer selection, as it is an ongoing challenge for service providers and consumers to select a trustworthy company.

This service negotiation framework in SDN introduces a pathway of building communication between service consumers and service providers that assists in developing a trust relationship between them. This experiment result demonstrates the success of the framework using artificial intelligence systems.

5.7 References

Baranwal, G., & Vidyarthi, D. P. (2014). A framework for selection of best cloud service provider using ranked voting method. 2014 IEEE international advance computing conference (IACC).

Chieng, D., Marshall, A., & Parr, G. (2005). SLA brokering and bandwidth reservation negotiation schemes for QoS-aware internet. *IEEE Transactions on Network and Service Management, 2*(1), 39-49.

Devi, R., Dhivya, R., & Shanmugalakshmi, R. (2016). Secured Service Provider selection methods in Cloud. 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS).

Fachrunnisa, O. (2011). *A methodology for maintaining trust in virtual environments*, Curtin University].

Fachrunnisa, O. (2016). A Framework for Service Formalization and Negotiation for Trust Maintenance in Digital Environments.

Garg, S. K., Versteeg, S., & Buyya, R. (2011). Smicloud: A framework for comparing and ranking cloud services. 2011 Fourth IEEE International Conference on Utility and Cloud Computing.

Gaurav Baranwal and Deo Prakash Vidyarthi. (2014, Feb. 2014.). A framework for selection of best cloud service provider using ranked voting method. Advance Computing Conf. (IACC),.

Gharakheili, H. H., Bass, J., Exton, L., & Sivaraman, V. (2014). Personalizing the home network experience using cloud-based SDN. Proceeding of IEEE International symposium on a world of wireless, mobile and multimedia networks 2014.

Ghosh, N., Ghosh, S. K., & Das, S. K. (2014). SelCSP: A framework to facilitate selection of cloud service providers. *IEEE transactions on cloud computing, 3*(1), 66-79.

Gomes, R. L., Bittencourt, L. F., & Madeira, E. R. (2013). SLA Renegotiation According to Traffic Demand. 2nd Workshop on Network Virtualization and Intelligence for the Future Internet (WNetVirt).

Gomes, R. L., Bittencourt, L. F., & Madeira, E. R. (2014). A similarity model for virtual networks negotiation. Proceedings of the 29th Annual ACM Symposium on Applied Computing.

Gomes, R. L., Madeira, E. R., & Bittencourt, L. F. (2017). Mechanisms for management of SLA for virtual software defined networks based on QoS classes. 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM).

Hussain, F. K. (2006). *A Methodology for Trust Management in Service Orientated*

Environment, Curtin University of Technology].

Jain, R., & Paul, S. (2013). Network virtualization and software defined networking for cloud computing: a survey. *IEEE Communications Magazine, 51*(11), 24-31.

Khan, S. (2020). *Guranteeing end-to-end QoS provisioning in SDN*. (FEIT, Issue. University of Technology Sydney.

Khan, S., & Hussain, F. K. (2020). Evaluation of SLA Negotiation for Personalized SDN Service Delivery. International Conference on Advanced Information Networking and Applications.

Körner, M., Stanik, A., & Kao, O. (2014). Applying QoS in Software Defined Networks by Using WS-Agreement. Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on.

Li, X., Ma, H., Zhou, F., & Yao, W. (2015). T-broker: A trust-aware service brokering scheme for multiple cloud collaborative services. *IEEE Transactions on Information Forensics and Security, 10*(7), 1402-1415.

Walayet Hussain, F. K. H., Omar Khadeer Hussain, EliZabeth Chang. (2016). Provider-Based Optimized Personalized Viable SLA(OPV-SLA) Framework to Prevent SLA violation. *The Computer Journal Advance Access, Section A*. (The British Computer Society)

Chapter 6

A SERVICE MANAGEMENT FRAMEWORK THAT EMPLOYS PROACTIVE AND PASSIVE CONTINUOUS SERVICE PERFORMANCE MONITORING

6.1 Introduction

The service level agreement (SLA) lifecycle-based framework is proposed in this thesis. The SLA framework is initiated with a trust-building stage that involves service negotiation and service requirements formalisation activities and is comprehensively discussed in chapter 5. The outcomes of the framework enable the service consumer and the service provider to negotiate the service requirements with a certain level of the services and formulate a service contract between the interacting parties by obtaining assistance from a third-party agent. It is also crucial to monitor the performance of the deliverable services to preserve the SLA contract as well as preserve the trust relationship between all interacting parties. Hence, we propose an SLA life-cycle-based framework with a service management framework that includes service performance monitoring assisted by a third-party agent.

This chapter proposes a service management framework comprising online performance monitoring approaches. Firstly, we develop a framework for proactive continuous performance monitoring that enables the third-party agent to a) continuously monitor the deliverable services and closely differentiate the service performance against the agreed degree of performance delivery, b) conduct counteractive measures to determine the service discrepancies from the agreed degree of deliverable services. Moreover, we implement early checkpoints and the SLA checkpoints in order to preserve the SLA with the intention of early intervention to preserve the SLA. Secondly, we develop a framework for passive performance monitoring that enables a third-party agent to analyse the run time traffic, determine the business demand and vulnerable traffic and assign priority rules based on the findings. As previously mentioned, this framework is the second stage of the SLA lifecycle framework.

To address the problem and recognise the importance of service management, we provide a case-based discussion in section 6.2 and articulate the issues. Then, we comprehensibly discuss the pro-active continuous performance monitoring approach in section 6.4.1 and the passive performance monitoring framework in section 6.4.2. Furthermore, we demonstrate a simulation-based proof of concept of the service management framework with results and discussions in section 6.5. Section 6.6 concludes the chapter.

6.2 Continuation of the ABC Education Pty Case Study and Problem Definition

ABC Education = service requester

SMART Mediation = third-party agent

XQuery Ltd. = service provider

Let us discuss a case study to understand the need to manage the services in a service-oriented software-defined network (SDN). In the previous chapter, we introduced ABC Education, a service requester or a service consumer who is looking for SDN services for their organisation. ABC Education has also defined 'n' number of service criteria and expectations. ABC Education requests assistance from a service mediation agent or third-party agent to find the service provider that would best fit this organisation. ABC Education wants to develop and maintain a healthy relationship with all interaction parties and maintain a long-term trust relationship. ABC Education has followed a Service Negotiation Framework to avoid dissatisfaction among service providers and consumer parties and develop a good relationship among all interaction parties. The service negotiation framework is discussed in detail in our previous chapter (Chapter 5. This framework assists ABC Education in intelligent decision-making to find a suitable service provider who is able to deliver their services according to their requirements. XQuery Ltd. is the service provider that ABC Education finds is the best fit to work with. On the other hand, XQuery Ltd. receives assistance in decision-making through the service negotiation framework and decides to deliver the services requested by the requester ABC Education.

SMART Mediation is an intermediate agent that assists collaboration and interaction between the service requester and providers. When all interaction parties are agreed, an SLA is drawn up among both parties, which provides information on all the agreed services and their agreed-upon expected reliability in a single document. The document clearly states the metrics, responsibilities, and expectations so that neither party can plead ignorance in the event of issues with the service. The types of SLA metrics required will depend on the services being provided.

Let us assume that XQuery Ltd. formally agreed to deliver network services to ABC Education according to an expected performance level of services. In the SLA, ABC Education lists various performance metrics to define the performance level. For example, service availability implies the time the network should be available for use (uptime). Service availability can be broken down further into time slots. For instance, ABC Education has a greater demand for uptime from 8:00 am to 7:00 pm. therefore, 99.5 % availability is required during uptime. The rest of the time, the availability should be more than 92%. After some time, ABC Education experienced a low quality of availability which included frequent drop-offs, and service was not always available during peak times. This situation continued for five weeks. ABC Education reported this to XQuery Ltd. as well as SMART mediation. SMART mediation independently investigated the issue raised by ABC Education. Their investigation revealed the following:

1. The following pattern of service unavailability for weeks 1 – 5 is shown in Table 6.1.
2. A service drop-off complaint was made on week five, and found the drop-off status below.

Week	Monday	Tuesday	Wednesday	Thursday	Friday
Week 1	1 hour	Good	1.5 hours	good	good
Week 2	2 hours	0.5 hour	good	good	good
Week 3	Good	good	0.5 hour	1 hour	good
Week 4	2 hours	good	good	good	0.5 hour
Week 5	Good	1 hour	1 hour	1 hour	1 hour

Table 6.1: Service unavailability status for five weeks.

The independent investigation identifies that the service dropped off and was unavailable beyond the expectation level defined in the SLA, and the issues are incrementally increasing. This means the number of service discrepancies deviates from the standard in the first and the following week; therefore, the number of service discrepancies is observed incrementally. This shows that the SLA has been violated, and thus penalty clause should be invoked. SLA penalties may lead to a mild or significant financial disadvantage for both parties and lead to

dissatisfaction, resulting in a very unreliable relationship. However, if the service performance is monitored continuously, possible service violations may be detected at an early stage and preserve the performance of the service that falls outside the parameters identified by the SLA.

Service management is a vital component that should be incorporated into the SLA to avoid the aforementioned situation. In other words, we can say that service management is a comprehensive continuous process to preserve the SLA. The service management elements should include the following; a) definitions of measurement standards and methods, b) reporting processes, content, and frequency, c) a dispute resolution process and an indemnification clause protecting the customer from third-party litigation resulting from service level breaches, and d) a mechanism for updating the agreement as required (Stephanie Overby, 2017). Service monitoring is a significant component of service management that assists in sustaining the quality of the service and sometimes detects any possible service degradation and takes practical actions.

From the above discussion, we can formulate the requirements for service management to align with customer demand and protect the SLA from being breached.

1. What kind of metrics should be monitored? This question should be addressed before implementing the service management process. The service monitoring metrics should be articulated, described, and discussed with all interacting parties. The final defined metrics should be aligned with the service requirements process that was discussed in our previous chapter(chapter 5).

2. From the above scenario, we recognise that deploying an approach to quickly detect performance discrepancies and resolve them is necessary. It is also essential to develop knowledge from the previous history of performance discrepancies and apply proper intervention processes to protect the service performance from sudden unexpected situations. Thus, an appropriate mechanism must be employed to address the above position and sustain the services.

3. To preserve service quality, it is often asked, "How often should we revise the SLAs?" There are several approaches available in the IT industry. However, it is wise to

implement a reliable monitoring approach that can predict a possible violation tendency as recognised by the monitoring agents or service providers and take action to prevent it. Therefore, the service is being preserved from actual SLA violations.

6.3 Background

Service management is a crucial component of an SLA. Service management requires continuous monitoring to ensure the contract conditions in the SLA are met. Therefore, continuous service monitoring plays a significant role in determining whether an SLA has been violated (Quillinan et al., 2010). Service monitoring is sometimes automated in SLA enforcement mechanisms at run-time without excessive delay (Quillinan et al., 2010). For example, in an SOA, once an SLA violation is determined and recorded, the agreed penalty may automatically be imposed. On the other hand, monitoring also facilitates traditional enforcement to avoid possible conflicts among interacting parties (Quillinan et al., 2010). For example, if any interacting parties challenge the imposition of the automatic penalty, a document that details the monitored data can verify whether the penalty is justified.

Monitoring an agreement necessitates sporadic testing to determine whether the agreement's terms and conditions have been met. This testing may involve a specific variable or a set of variables, such as network failover recovery or categorisation communication between the service receiver and provider according to the terms and conditions of the agreement. Furthermore, monitoring intervals and duration need to be specified and appropriately executed depending on the nature of the agreement. Monitoring can be of various types, such as complex or straightforward monitoring using several benchmarking formulas upon agreement (Quillinan et al., 2010). For example, monitoring may occur based on a single feature or parameter of the network, such as "host is available." Other monitoring may involve an evaluation or benchmark after measuring a set of parameters in the network, for example, "host availability is 3% more than the defined threshold".

A possible violation can be identified through online monitoring or post facto auditing of the service performed (Quillinan et al., 2010). In terms of auditing, a definitive decision must be made considering accurate monitoring logs. In contrast, online monitoring is the pre-auditing stage that needs to be performed accurately.

As discussed in Chapter 4, the service management framework is the second module of the QoS-guaranteed SLA lifecycle framework. Service management is coordinated by a special agent or network management and violation detection agent (neutral position). This module will monitor the run-time network and gather updated network status information. The outcome of this module is an online monitoring log. Design details are provided later.

Our previous chapter presented the service negotiation and SLA formulation framework in SDN. This chapter presents a service management framework where monitoring all the network services is one of the primary elements of the SLA management framework.

Continuous service monitoring tasks can be observed in two ways: online monitoring and post facto auditing. We have viewed SDN as a service-oriented principle and developed our frameworks to deliver network-as-a-service. The online monitoring approach is chosen to perform continuous monitoring tasks. This can be performed by either a trusted third-party or trusted module at the service provider site or a trusted module on the customer site. The details of the online monitoring approach are discussed in the following section.

6.3.1 Online Monitoring

Online monitoring is a commonly used monitoring approach for SLAs between interacting parties. Details of SLA construction are discussed in Chapter 5.

As online monitoring is performed in real-time, it is imperative to secure monitoring updates to protect malicious parties. Several commonly used online monitoring techniques (Keller & Ludwig, 2003). For our framework, we choose both proactive and passive service monitoring

approaches to perform our network monitoring to manage the services and preserve the SLA. A proactive monitoring approach is an advanced form of active monitoring by troubleshooting the network to identify potential problems before impacting the end user's performance(Fachrunnisa, 2011). We choose proactive monitoring for this framework, as we intend to perform run-time monitoring of the network and determine any potential issue before impacting the service consumer. In addition, we are not limiting our monitoring approach to run-time monitoring only. We also use passive performance monitoring that leverages the opportunities of analysing and releasing the results from run-time data captured from run-time monitoring and inspecting the data to gather the current network status and make possible adjustments to the deliverable services if needed.

6.3.1.1 Passive Network Monitoring

Passive monitoring gathers network traffic data and analyses it over a specific period. The monitor then studies the analysis and releases the results. Passive monitoring generally studies the ransom status of the network and analyses real-time traffic data from specific points in the network rather than analysing test data. Passive network monitoring is the process of capturing network traffic and inspecting it closely to gather information on the current traffic status. The proactive monitoring configuration is determined and configured based on the current traffic status.

6.3.1.2 Proactive Network Monitoring

A proactive monitoring approach is an advanced form of active monitoring by troubleshooting the network to identify potential problems before impacting the end user's performance(Fachrunnisa, 2011). This approach performs continuous monitoring to search for any indication of issues or any possible issues that are about to occur. This monitoring approach allows the monitoring agent to uncover problems that might radically affect the network, stop them, and keep the network on track. Introducing proactive network monitoring can leverage the enterprise by protecting the network by automatically discovering any network issues at any time. It also scans the network for problems and provides real-time alerts if something is

out of the ordinary. The most significant benefit of proactive monitoring is finding the malware hiding in the network.

Passive and active monitoring are both critical in their way. Passive monitoring reveals the end-user's perspective using real-time performance data, whereas proactive monitoring generates data to determine potential network issues that may violate the SLA and maintain visibility. A combination of both approaches is the best way to monitor network performance and modify the network accordingly.

6.4 Service Monitoring Framework for Service Management

This section presents a service monitoring framework that monitors services to ensure the performance meets the SLA standard. As previously discussed, online monitoring is a common way to monitor the services and is undertaken by a trusted third party and can be performed by a trusted module on the service provider side or a trusted module on the consumer side.

It has been found that the proactive, continuous monitoring approach is able to maintain trust, as this approach enables "building the commitment and examining very often" (Fachrunnisa, 2011). Therefore, a well-established trust can be examined often to keep it constant using this approach(Fachrunnisa, 2011). On the other hand, passive monitoring is able to provide a clear understanding of a defined guideline about how to construct a proactive monitoring plan in terms of circumstances and services. Therefore, we argue that it is wise to implement a combined approach in this framework.

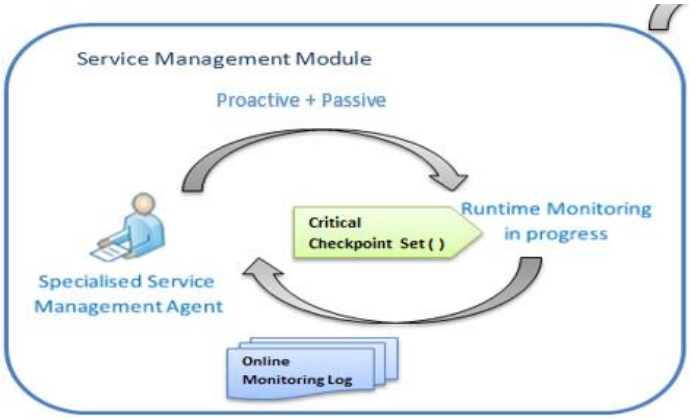


Figure 6.1: *The service management module consists of proactive and passive monitoring* by S. Khan, 2022

Figure 6.1 shows a high-level overview of service monitoring for the service management framework. A specialised service management agent regularly monitors the delivered services to ensure the quality of the services delivered to the consumer side.

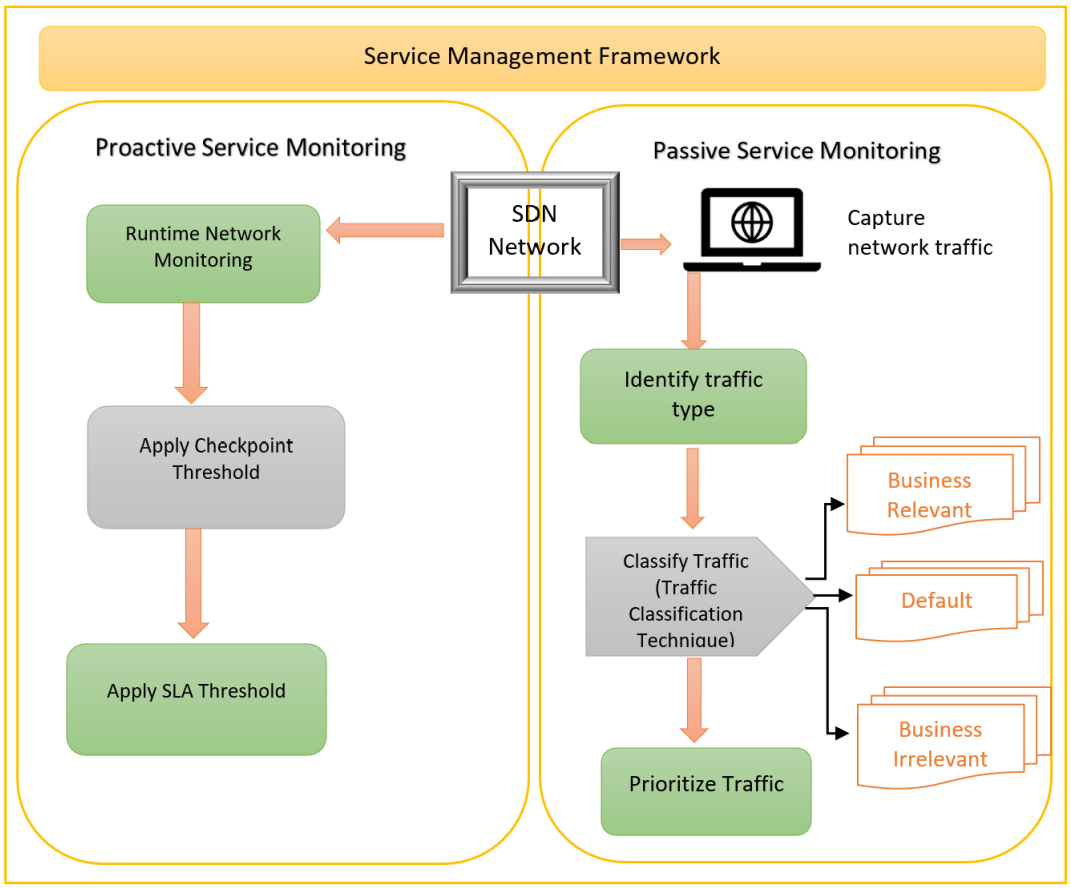


Figure 6.2: *A detailed overview of the proactive and passive monitoring framework* by S. Khan, 2022.

We include two types of monitoring approaches agreed upon in the SLA performed by the agent. The passive monitoring approach helps the agent collect information about the real-time network, such as a) the status of the network, b) the type of traffic passing through the network, and c) which traffic should be prioritised. The agent collects a sample of traffic from a run-time network for a specific duration of time. On the other hand, the proactive monitoring approach allows various checkpoints in the QoS of the network to be applied. The checkpoint allocation may vary depending on the knowledge gathered from the passive network monitoring statistics. The detailed framework is shown in Figure 6.2. The approach is iterative; in other words, both monitoring approaches depend on each other. Proactive monitoring performs the real-time monitoring of the network by applying checkpoints where these checkpoints may vary depending on the statistical analysis of the traffic data collected using the passive monitoring approach.

This section presents our service monitoring framework for managing the delivered services. Firstly, we discuss the proactive, continuous performance monitoring approach for real-time performance monitoring. Secondly, we discuss passive performance monitoring for analysing the network traffic and developing the monitoring rule that can be applied in proactive performance monitoring.

6.4.1 Proactive Continuous Performance Monitoring Framework

As discussed in section 3, proactive monitoring enables immediate action to be taken when a violation or complaint occurs. Similarly, pro-active continuous performance monitoring examines the network to observe whether the network's performance does not exceed the checkpoint threshold. The objective of this framework is to enable the service provider to proactively examine network services to avoid SLA violations and prevent future possibilities

of SLA violations. Early intervention is crucial to sustaining the SLA as it involves many obligations and penalties.

To achieve the defined objective, we propose the following mechanisms for conducting proactive continuous performance monitoring:

1. Proactively monitor the network using the run-time monitoring approach.
2. Determine and implement the early checkpoint threshold according to the SLA.
3. Determine and implement the SLA threshold according to the SLA.

To realise the process of proactive continuous performance monitoring, let us re-examine the case study discussed in section 2. A close examination of Table 6.1 shows the different performance of the services as follows: in the first week, the services were unavailable for 1 and a half hours; in the second week, the services were unavailable for 2 and a half hours; in the third week, the services were unavailable for 1 and a half hours; in the fourth week, the services were unavailable for 3 hours; and in the fifth week, the services were unavailable for 4 hours. The scheduled one-off monitoring session identified that the services were unavailable for 12 and half hours over the five weeks, and it also identified that the service quality degraded progressively every week. After reviewing the SLA, the service monitoring agent identifies that the services must be available 99.6% of the time, which means the services must be available 478.08 hours out of 480 hours over the five weeks. Hence, the monitoring agent identifies that the service unavailability should not exceed 2 hours over five weeks; however, the scheduled one-off monitoring reveals that the services were unavailable for 12 hours and half hours over five weeks, and the service quality degrades progressively. As no continuous performance monitoring or real-time monitoring is undertaken each week, there is a significant possibility that the service performance will fall outside the parameters defined by the SLA.

One-off performance evaluation is undertaken at the end of the interaction if the services have not been delivered as mutually agreed upon by the two parties. One of the significant weaknesses of one-off performance evaluation is that if the performance in week one degrades by week two and so on, this is not detected until performance evaluation occurs. Over time, minor performance discrepancies will most likely continue to grow, leading to a serious crisis

because these discrepancies were not resolved earlier. Moreover, this situation leads to an increased challenge to restore everything with every given week, thereby experiencing performance diminishing. Hence, the above discussions reflect the need for proactive continuous performance monitoring rather than reactive performance monitoring.

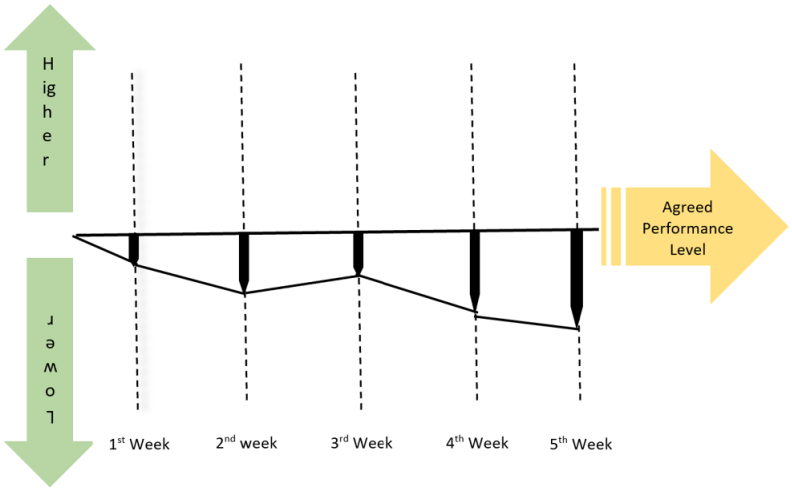


Figure 6.3: One-off performance evaluation reflects service unavailability over the five-week duration. Adapted from *One-off performance evaluation*, by Fachrunnisa, 2011.

The proactive continuous performance monitoring approach opens up the possibility of eliminating any partial effect from any parties. The proactive continuous performance monitoring approach examines service performance at every assigned checkpoint, which reduces the possibility of the performance degrading over time. Therefore, intervention can be undertaken at an early stage. In addition, introducing proactive continuous performance monitoring reduces the risk of SLA failure because it ensures that a service will be delivered effectively as agreed (Fachrunnisa, 2011). Figure 6.3 reflects the pattern of the performance discrepancies over five weeks since a performance assessment is undertaken at the end of the interaction. Figure 6.4 illustrates that proactive performance continuous monitoring proposes strategies to minimise performance discrepancies by applying proper checkpoints during the interaction.

We propose using run-time performance monitoring and applying checkpoints to monitor the SDN's performance continuously. As trust is established in executing agreements or mutually

agreed-on behaviours (Fachrunnisa, 2011), we argue that the proposed proactive continuous performance monitoring provisions the trust relationship between service consumers and providers. In addition, in the case of unmet performance or an agreement not being fulfilled, the intermediate agent continuously monitors the services to predict any possible violation and assists in the early detection of potential service-level agreement violations. Therefore, a pre-determined performance threshold must be established.

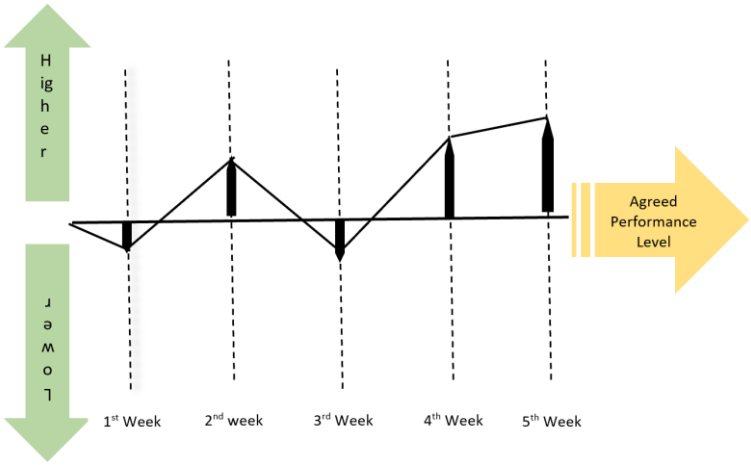


Figure 6.4: Proactive continuous performance monitoring reflects the performance of the services over the five weeks. Adapted from *Proactive continuous performance monitoring*, by Fachrunnisa, 2011.

This section illustrates the step-by-step process of conducting a proactive monitoring framework. The framework comprises three steps carried out by the third-party agent, where both parties agree upon the monitoring parameters. Consequently, both interacting parties need to agree with the individual who will act as a third-party agent.

Step 1: Proactively Monitor the Network Using the Run-time Monitoring Approach.

The first step of the framework includes an activity to perform a real-time monitoring approach of the network and to determine the checkpoints that need to be deployed in the monitoring approach. Besides, The primary objective of this framework is to minimise any possibility of

violating the SLA performance commitment. We further define the objective into two sub-objectives to achieve the broader objective. The first sub-objective is implementing a real-time monitoring approach to monitor network performance continuously. The second sub-objective is to ensure proactive action is taken if any event or alert occurs.

To achieve the first sub-objective, we simulate the SDN using simulation software. Several open-source network monitoring software packages are available to monitor the network's performance; most are highly customisable. Subsequently, we develop a virtual machine for the network monitoring server and integrate the server with the simulated network. Thus, we ensure that the network performance monitoring service can communicate with every network component. The monitoring service includes several features to customise the monitoring according to predefined parameters.

To achieve the second sub-objective, we introduce a proactive monitoring approach. In this approach, we apply the early performance degradation checkpoint threshold to identify the performance degradation and apply an immediate action policy to eliminate the possibility of compromising the SLA. Similar to the real-time monitoring approach, the proactive monitoring approach is a continuous process, and the intermediate agent encourages setting up an early performance degradation checkpoint alert. The details of the early detection and SLA checkpoints are discussed in the next section.

Step 2: Determine and Implement the Early Checkpoint Threshold According to the SLA.

Early detection checkpoints are the performance discrepancy threshold employed to determine service performance discrepancies. Besides, it is also essential to determine the number of checkpoints that need to employ to check the performance. Therefore, In order to determine the performance discrepancy information in a run-time environment, we applied the service performance discrepancies threshold in the run-time system. The intermediate agent determines the number of checkpoints and discrepancy thresholds during the SLA formulation

stage to inform all interacting parties and provide the appropriate documentation. Thus, the number of checkpoints and service discrepancy threshold will vary depending on the scenario.

Let us consider that to detect performance degradation early, and the monitoring agent applies the early detection checkpoint threshold of 20. As previously discussed, an early checkpoint threshold is applied in the run-time monitoring system for early intervention before the service performance degradation reaches such a level and violates the SLA. Therefore, in this approach, if the performance discrepancy level reaches the early degradation checkpoint level, the network monitoring system alerts indicate that immediate attention is required. The triggering alert may lead to various proactive actions taking place. The implementation details of the early detection checkpoints are discussed in the implementation and validation section.

Step 3: Determine and Implement the SLA Threshold According to the SLA.

The SLA threshold checkpoints determine the performance discrepancy threshold of services and the number of points that are necessary to check to determine the performance and identify any performance discrepancies. The intermediate agent needs to determine the number of checkpoints and the discrepancy thresholds during the SLA)formulation stage to inform all the interacting parties and provide the appropriate documentation. Thus, the number of checkpoints and the service discrepancy threshold will vary depending on different scenarios.

6.4.2 Passive Performance Monitoring Framework

This section details the step-by-step process of the passive performance monitoring framework. The framework consists of four steps to be overseen by the third party or service management agent. As we know from the above discussion, all interacting parties agree on who will act as a third-party agent. The agent's role is to manage the performance quality neutrally and take appropriate actions when required. Figure 6.5 illustrates passive monitoring, which consists of four steps. Each step is interconnected, which means that the outcome of the previous step is the input for the next step. The details of the steps are as follows.

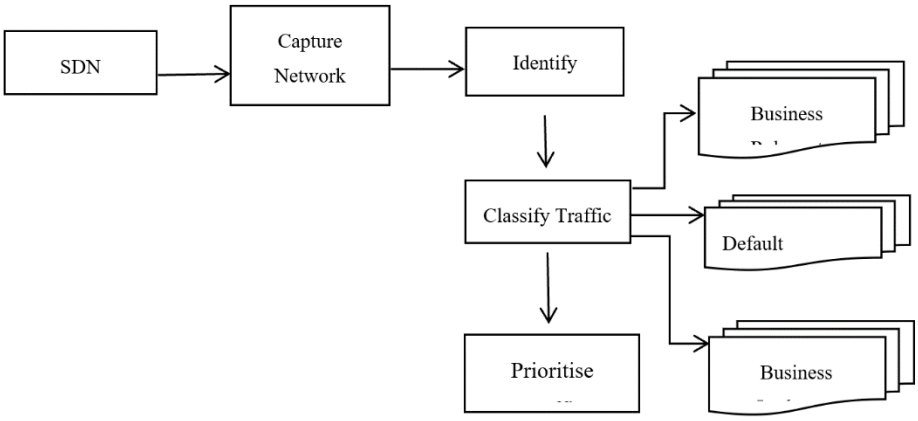


Figure 6.5: *Passive performance monitoring* by S. Khan, 2022.

Step 1: Capture Network Traffic

Figure 6.5 overviews the proposed passive performance monitoring framework. The framework captures the run-time traffic data according to predefined time slots. The extraction of the real-time traffic data includes all the services currently running on the network. In addition, the real-time traffic data gives an overall understanding of the status of the current network. As discussed in the previous section, extracting real-time network traffic allows us to analyse the network's overall performance data and determine whether all the services running are committed to the SLA.

This is a time-intermission-based continuous process to determine if any early intervention is required by making a reasonable adjustment. Therefore, the network traffic capture should occur according to the predefined time slot agreed upon by all the interacting parties and identified in the SLA. Determining the time window is a crucial task that includes two elements; Time space and time slot, and the concept are inspired by the work (Fachrunnisa, 2011) in relation to the framework for proactive continuous performance monitoring.

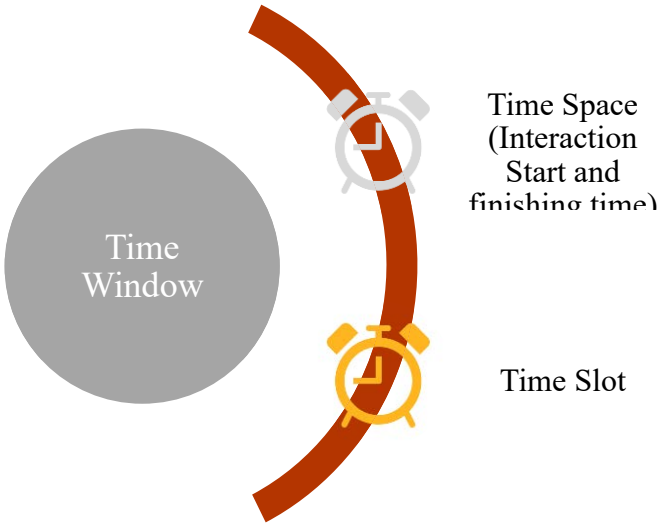


Figure 6.6: *Time window, which consists of time space and the time slot* by S. Khan, 2022.

Time Window of Interaction:

The time window of interaction is "a set of time statements that *represent when the context of the interaction is in question"* (Chang et al., 2006). Thus, the time window (Figure 6.6) of interaction can be described as an interval in time during which an interaction must take place. In our framework, a time window is a period when the third-party agent actively interacts with the systems and obtains their required information. In this framework, the time window of interaction is a predefined process of SLA and may perform iteratively.

Time-Space of Interaction:

The time-space of interaction is described as "*the total duration of time over which the behaviours of the trusted agent will be analysed, and trustworthiness measure and prediction will be carried out*." (Chang et al., 2006). The time-space of interaction is a predefined process that is stated in SLA and has been mutually agreed upon by both parties. According to the framework, the time-space entails interacting with the system's starting and finishing time, as stated in the SLA.

Time Slot of Interaction:

A time slot of interaction is defined as a '*finite number of non-overlapping, mutually exclusive and equally spaced sectors of time' (Chang et al., 2006)*. We define it as a series of an equal length of time allocated for a particular activity. According to our framework, the time slot of interaction refers to an allocated period assigned for an event, such as capturing the real-time network traffic of a running network.

Step 2: Identify Network Traffic Type

Identifying the network traffic type from a set of captured traffic is the second step of the passive performance monitoring framework. The basic idea is to identify the traffic type, such as VoIP traffic, network control traffic, video conference traffic, and data transmission traffic from the unknown traffic set captured in the previous step (section 4.1.1). We used the deep packet inspection-based method that uses current flow statistical features to determine the traffic type. Therefore, a machine learning-based algorithm (support vector machine) is used in artificial intelligence.

A support vector machine is a supervised learning model associated with learning algorithms to analyse classification and regression data. The technical approach we have followed is the same as most supervised machine learning models. We train a machine learning model by applying the dataset with the correct classification associated with them. Therefore, the supervised, trained learning model can successfully predict the new dataset's classification.

We use a software named Rapid Miner to implement the machine learning technique, and we discuss the implementation details in the implementation section.

Step 3: Classify Network Traffic

After successfully identifying the traffic type from a set of unknown traffic, the third step of the passive network monitoring framework is to classify the identified traffic. Classifying network traffic is an approach that allows organising the traffic into traffic classes or categories according to whether the traffic matches specific criteria.

We have categorised the traffic type identified from our used dataset according to the applications or services at this stage. We classify the traffic in three steps, as detailed in Table 6.3, which shows the three levels of traffic classification used in the framework.

Step 1: In the first step, we identify and list 16 traffic categories available from the dataset in the 3rd column. Cisco also classifies all traffic types into ten categories listed in the 4th column of the table. The Cisco classification table is presented in Appendix A. To enhance the framework's usage with an open scope, we add the Cisco classification categories with our identified classification categories according to the closest applications or services match.

Main Category	Sub-category	Traffic Category type from the dataset	Traffic category types from Cisco
Business relevant	**Network and Control Plane Protocol**	Network	Network Control
		VPN	
		Cloud	Operations/ Administration/ Management (OAM)
	Voice Application	VoIP	Voice
			Signaling
	Video Application	Streaming	Multimedia Streaming
		Video	Multimedia Conferencing
			Real-time Interactive
			Broadcast Video
	Data Application	Email	Transactional Data
		Collaboration	
		Web	
Default	**Default Forwarding**	Software Update	Bulk Data
		Download-File Transfer-File Sharing	
		System	
Business Irrelevant	**Business Irrelevant Category**	Music	
		Media	
		Social Network	
		Shopping	

Table 6.3: Traffic Classification Categories

Step 2: In the second step, we identify six broad sub-categories where all traffic categories or, in other words, service traffic categories can fit in. The newly identified sub-categories listed in Table 6.3 are detailed as follows:

1. **Network and Control Plane Protocol**: The Network and Control plane protocol category consists of network control and management traffic, virtual private network (VPN) traffic, and cloud services traffic. From Cisco's proposed category, we comprised the network control traffic and operations/ administration/ management (OAM) traffic in this category.

2. **Voice Application:** We precisely identify VoIP traffic from our analysed dataset for the Voice Application category. Additionally, we include all voice applications in this category defined by the Cisco ToS classification approach.

3. **Video Application:** The Video Application traffic category is straightforward, and our detected streaming and video traffic are in this category. Similarly, multimedia streaming, multimedia conferencing, and real-time interactive broadcast video services are in this category.

4. **Data Application**: Data Application is another critical category where email, collaboration, and web services are located. Similarly, business transactional data services from Cisco are incorporated in this category.

5. **Default Forwarding:** From our findings from the dataset, the Default Forwarding category consists of software update services, file download, file transfer, and file-sharing services traffic. Cisco advised that bulk data transaction services take place in this category.

6. **Business Irrelevant:** The Business Irrelevant category is the least essential category and assumes that the business has very little or no dependency in this category. Music, media, social networking services, and shopping are in this category.

Step 3: In the third step, we classify the associated sub-categories according to business demand as the top category in Table 6.3. We find three significant categories. We illustrate the network traffic classification approach as a hierarchical approach in Figure 6.7.

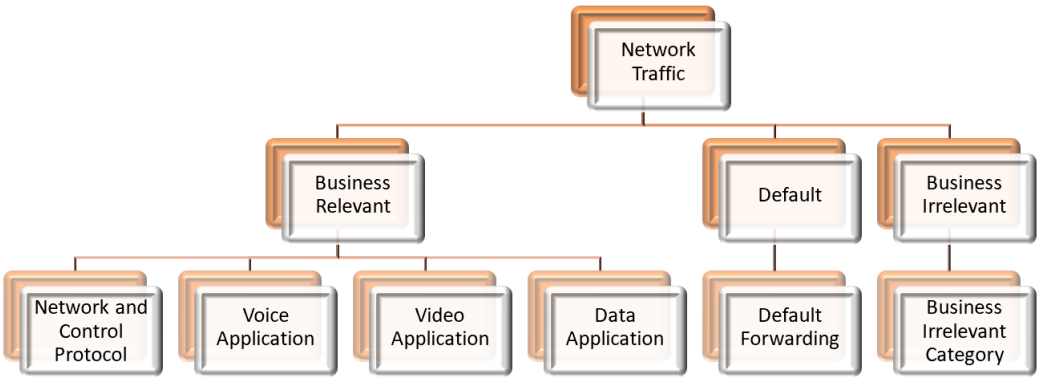


Figure 6.7: *A hierarchical approach to traffic classification* by S. Khan, 2022.

A) ***Business Relevant Category*** is the most important category where all the related services and applications directly support the business objectives and are necessary to operate a business or meet the business demand. This service group refers to a "no compromise" service group where service interruptions are regarded as deplorable. Thus, the applications of this category should be of significant importance and considered according to business best-practice endorsements.

B) ***Default*** is the second-most important category that lies in the services that may or may not support business objectives and are essential to operate the business without any interruption. In other words, the default category is the service support group category. Applications of this category should be considered with a default forwarding service.

C) **Business Irrelevant category** is the last and least important service category, which means the applications do not support business objectives and are typically consumer-oriented. Applications in this category should be given less priority.

Step 4: Prioritise Network Traffic

Traffic prioritisation is a method to add a traffic value to ensure that essential or time-critical traffic flows without delays and assists in service differentiation. Moreover, prioritising the network traffic helps sustain the QoS in the network (Cisco Systems, 2017). The network traffic prioritisation method is primarily required to guarantee a certain QoS to meet the SLA. We used the traffic prioritisation concept in this framework and developed our method to deliver differentiated services in the SDN services platform. To propose our traffic prioritisation approach, we were involved in deep research on various traffic prioritisation and differentiation methods. Of these, Cisco IOS Release is one of the significant methods.

Cisco IOS Release uses three bits for IP precedence in the Type of Services (ToS) field of the IP header and partitions the traffic into a maximum of 6 classes. Each precedence corresponds to a name defined as a "continue to evolve" and defined in RFC 719 (Cisco Systems, 2013). The table is shown in Appendix A. Following Cisco's method, we prioritised our pre-categorised traffic using the following approach shown in Table 6.4. Based on the urgency and dependency to sustain the QoS, we prioritised the sub-category in the following manner:

Main Category	Sub-category	Priority Number
Business Relevant	**Network and Control-Plane protocol**	1
	Voice Application	2
	Video (Interactive)	3
	Video (Streaming)	3
	Data Application	4
Default	**Default Forwarding**	5
Business Irrelevant Category	**Business Irrelevant Category**	6

Table 6.4 Demonstrates the network traffic prioritisation categories identified in our research.

In the previous section, we discussed the three primary traffic categories. Therefore, we further identify six sub-categories. Based on our identification, business-relevant categories received the top priority compared to the other categories. Let us discuss the traffic sub-category and prioritisation reasoning in 4.1.3, and we discuss the three broad categories in the following.

1) *Network and Control Plane Protocol*: This class is for network control and traffic management and ensures consistent and reliable network operations. Therefore, this traffic class is critical and guarantees bandwidth queue traffic. The defined class implies the queue should be provisioned with moderate but dedicated traffic and not dropped. Thus, no early detection alert is expected to be positioned. On the other hand, network operations, administration, and management traffic can be characterised as a critical traffic group essential for ongoing network maintenance and support.

   The traffic type that incorporates this traffic group is as follows:

   - Network routing traffic includes EIGRP, OSPF, BGP, OpenFlow, HSRP, and IKE.
   - Operations/ Administration/ Management (OAM) traffic includes SNMP, SSH, Syslog etc.
   - Network control and management traffic
   - Virtual Private Network (VPN) traffic,
   - Cloud services traffic.

2) *Voice Application*: This class is for voice audio traffic and should be provisioned with a strict priority queuing service; hence, it is necessary to control admission to this class. All audio components of multimedia conferencing applications remain in this class. The application traffic in this class is Cisco Jabbar, Teams, Webex, and Spark.

3) *Video Application:* Similar to the above, this class is for video traffic and has unique QoS requirements depending on the type of video. Various video applications generate various video traffic. For simplicity, we keep all video traffic within the same class. However, according to business demand, it is possible to categorise this further and prioritise them. The possible video traffic that may be generated is as follows:

   a. Broadcast Video: This traffic class is for broadcast TV, live events, video surveillance flows, and similar streaming video flows.

b. Real-Time Interactive Video: This traffic class consists of an inelastic interactive video application. Admission to this class should be controlled. An example of real-time interactive traffic is Cisco telepresence traffic.

c. Multimedia Conferencing Video: This traffic class carries an elastic interactive multimedia collaboration application. The traffic in this group should be provisioned with a guaranteed bandwidth queue with differentiated service code points. Example applications of this type of traffic include Cisco Jabber, Webex, and Spark.

d. Multimedia Streaming: This traffic class carries elastic streaming video applications, such as video-on-demand(VoD). Traffic in this class should be provisioned with a guaranteed bandwidth queue. Examples of multimedia streaming traffic are Cisco Digital Media System VoD streams and E-learning videos.

4) *Data Application:* Data application consists of two categories of applications, foreground and background applications. Users expect a response via the network to continue with their tasks, referred to as *foreground* applications. Excessive latency to such applications directly impacts user productivity. Conversely, users do not directly impact their productivity though they are business relevant and typically consist of machine-to-machine flows referred to as *background* applications.

We know the data application class consists of email, collaboration and web category traffic from table 6.3 above. This can be referred to as transactional data traffic. This traffic is from interaction and foreground data applications. Therefore, traffic in this group should be provisioned with a dedicated bandwidth queue. Examples of the traffic in this class include data components of multimedia collaboration applications, enterprise resource planning applications, customer relationship management applications, and database applications.

5) *Default:* The default class consists of the traffic generated from default application forwarding. The traffic in this class is for non-interactive *background* data applications and should be provisioned with a dedicated bandwidth queue. An example application of this class includes email, backup operations, FTP/SFTP transfers, video, and content distribution.

6) *Business Irrelevant Category:* This class carries the traffic from those applications that are identified as business irrelevant. The traffic in this class is permitted on business networks when bandwidth is available and should be provisioned with a minimal bandwidth queue. Examples of this traffic include YouTube, iTunes, and Bit Torrent.

Four categories of traffic fall in the business category, and as mentioned above, we prioritise them based on urgency and dependency within the network.

6.5 Experiments and Results

The previous section presents the service management framework, an approach or element to managing or sustaining the SLA; therefore, successfully implementing the proposed framework initiates a trust relationship. The framework comprises two monitoring approaches: passive monitoring and proactive continuous monitoring approaches. Each approach has been developed and validated through implementation, which enables it to integrate with our other proposed frameworks, as detailed in chapters 7 and 8, to validate the success of the SLA-based QoS guaranteed framework in service-oriented SDN. To implement the aforementioned frameworks, we engineered the systems using the following tools.

1. Wireshark: A free and open-source packet analyser. It is primarily used to analyse traffic, communication protocol, network troubleshooting, and software development and protocol development.
2. Rapid Miner Studio Version 9.6
3. Support Vector Machine model
4. Python

5. Virtual Box
6. GNS3
7. Zabbix

This section describes the development of the Cisco network prototype using GNS3. Since we know, in terms of QoS traffic, there is no difference between Cisco performance traffic data and SDN performance traffic data(Braun, 2016). Therefore, to reduce the complexity of the implementation, we developed a Cisco network prototype and continued our experiments.

The following section discusses the implementation process of the service management framework that includes a passive performance monitoring approach and a proactive performance monitoring approach and demonstrates the implementation results, followed by a discussion of the framework. The implementation details of the approaches are described in the following section.

6.5.1 Implementation of Proactive Performance Monitoring:

The proactive continuous performance monitoring framework is a monitoring approach that enables run-time network monitoring and taking proactive actions when required. The details of the framework are described in section 6.4 of this chapter. Since the objective of the framework is to monitor the network performance in a real-time manner, we set up a real-time network monitoring environment by simulating the network environment with a run-time monitoring tool that performs network monitoring in real-time and applying proactive action features according to the monitoring status.

Step 1: Proactively Monitor the Network Using the Run-time Monitoring Approach

The first step of proactive performance monitoring is to develop a simulated network, and for this purpose, we use the gns3 simulation and emulation tool. To monitor the simulated network

in run-time, we deploy a run-time monitoring agent in the network. We configured the Zabbix virtual machine and integrated the virtual machine (VM) into our developed network simulation to ensure the Zabbix appliance can perform real-time monitoring of the whole network. The network simulation is developed following the network design demonstrated in Figure 6.8. The prototype development process is described in Appendix A. The network architecture design shows that three sites, named Summer, Spring, and Fall, are interconnected using a switch. Zabbix appliance 6.0 VM relates to a switch and ensures that all the devices can communicate with Zabbix VM simultaneously. Figure 6.8 illustrates that all the devices are active and communicate without disruptions.

The Zabbix monitoring tool monitors the network using a browser with a pre-setup IP address. Moreover, browser-oriented monitoring facilitates access to the monitoring front end or dashboard from a remote location.

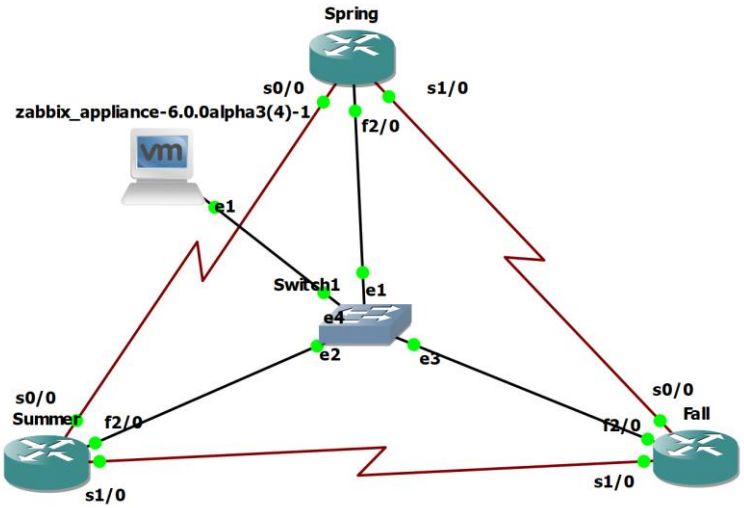
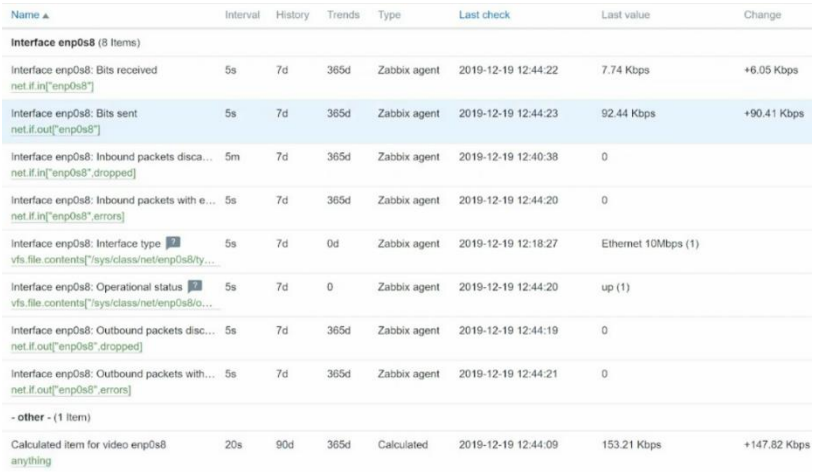


Figure 6.8: *The developed network simulation and architecture design* by S. Khan, 2022.

Step 2: Determine and Implement the Early Checkpoint Threshold According to the SLA

Determining and deploying the early checkpoint threshold according to the SLA is critical in the proactive continuous performance monitoring framework. The details of the process are discussed in section 6.4.1.



Name ▲	Interval	History	Trends	Type	Last check	Last value	Change
Interface enp0s8 (8 Items)							
Interface enp0s8: Bits received net.if.in["enp0s8"]	5s	7d	365d	Zabbix agent	2019-12-19 12:44:22	7.74 Kbps	+6.05 Kbps
Interface enp0s8: Bits sent net.if.out["enp0s8"]	5s	7d	365d	Zabbix agent	2019-12-19 12:44:23	92.44 Kbps	+90.41 Kbps
Interface enp0s8: Inbound packets disca... net.if.in["enp0s8",dropped]	5m	7d	365d	Zabbix agent	2019-12-19 12:40:38	0	
Interface enp0s8: Inbound packets with e... net.if.in["enp0s8",errors]	5s	7d	365d	Zabbix agent	2019-12-19 12:44:20	0	
Interface enp0s8: Interface type vfs.file.contents["/sys/class/net/enp0s8/ty...	5s	7d	0d	Zabbix agent	2019-12-19 12:18:27	Ethernet 10Mbps (1)	
Interface enp0s8: Operational status vfs.file.contents["/sys/class/net/enp0s8/o...	5s	7d	0	Zabbix agent	2019-12-19 12:44:20	up (1)	
Interface enp0s8: Outbound packets disc... net.if.out["enp0s8",dropped]	5s	7d	365d	Zabbix agent	2019-12-19 12:44:19	0	
Interface enp0s8: Outbound packets with... net.if.out["enp0s8",errors]	5s	7d	365d	Zabbix agent	2019-12-19 12:44:21	0	
- other - (1 Item)							
Calculated item for video enp0s8 anything	20s	90d	365d	Calculated	2019-12-19 12:44:09	153.21 Kbps	+147.82 Kbps

Figure 6.9: *List of the items in the Zabbix monitoring dashboard* by S. Khan, 2022.

Firstly, we configure the item according to the trigger or alert we apply for the early checkpoint. The item collects the network's information according to the trigger's need. Figure 6.9 shows the list of the sample items that we used for our proactive performance monitoring framework.

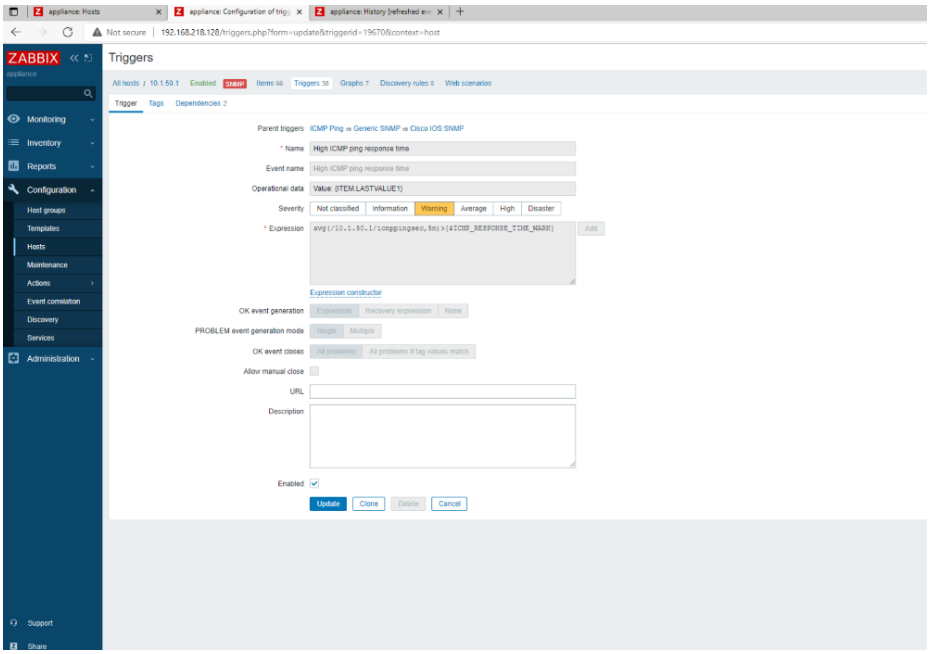


Figure 6.10*: Trigger configuration using the Zabbix dashboard* by S. Khan, 2022.

Secondly, we set up an alert to identify any performance degradation. The alert is configured to compare the current network performance with a predefined performance threshold. The checkpoint alert will trigger if the performance begins to degrade and exceed the threshold. We set up triggers with various severity levels.

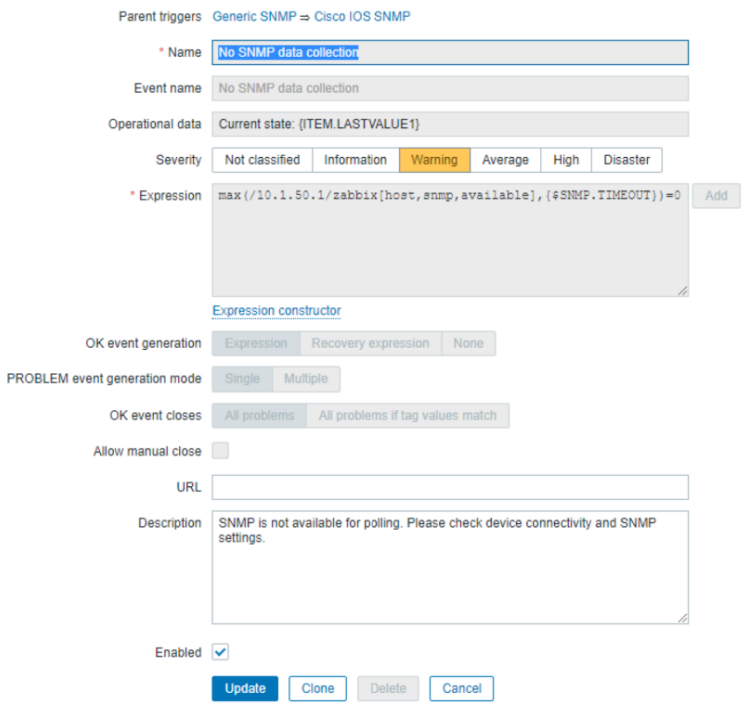


Figure 6.11: *Checkpoint trigger configuration detail for early detection* by S. Khan, 2022.

The alert will be triggered or activated based on the network's performance decompose level and impact level. Thus, the monitoring agent can identify the triggered alert's severity level status. Figure 6.10 shows the triggers we set up for our experiment, and Figure 6.11 shows the detail of the trigger configuration and the level of severity that can be employed.

Step 3: Determine and Implement the SLA Threshold According to the SLA.

Identifying and applying the SLA threshold checkpoint is a significant step in the proactive continuous performance monitoring framework. Section 6.4.1 discusses the SLA threshold

checkpoint approach. In this step, we share the items configured in the previous section and apply the trigger or alert for the SLA checkpoint.

We know from the previous section that items collect the network's required information according to the checkpoints and SLA alerts. Figure 6.9 lists the items we coordinated and used for our monitoring framework. In this step, we create the SLA threshold alert. The SLA threshold for each service should be pre-determined and documented in the SLA contract. A similar process is followed to apply the SLA checkpoints that we have applied to configure the early detection checkpoint. The only difference between these two approaches is that applying SLA checkpoints is more critical than applying early detection checkpoints. Two sample SLA checkpoints are shown in Figures 6.12 and 6.13.

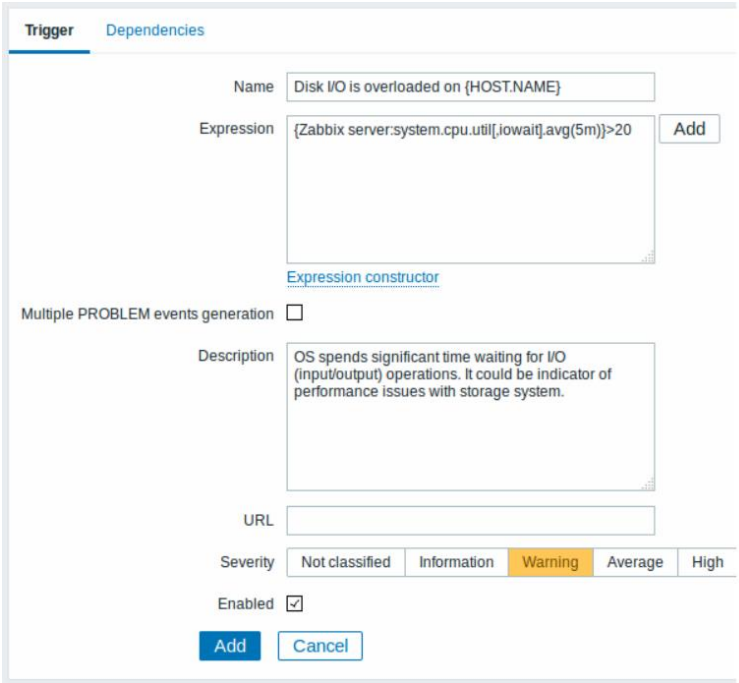


Figure 6.12: *SLA checkpoint in the Zabbix monitoring dashboard* by S. Khan, 2022.

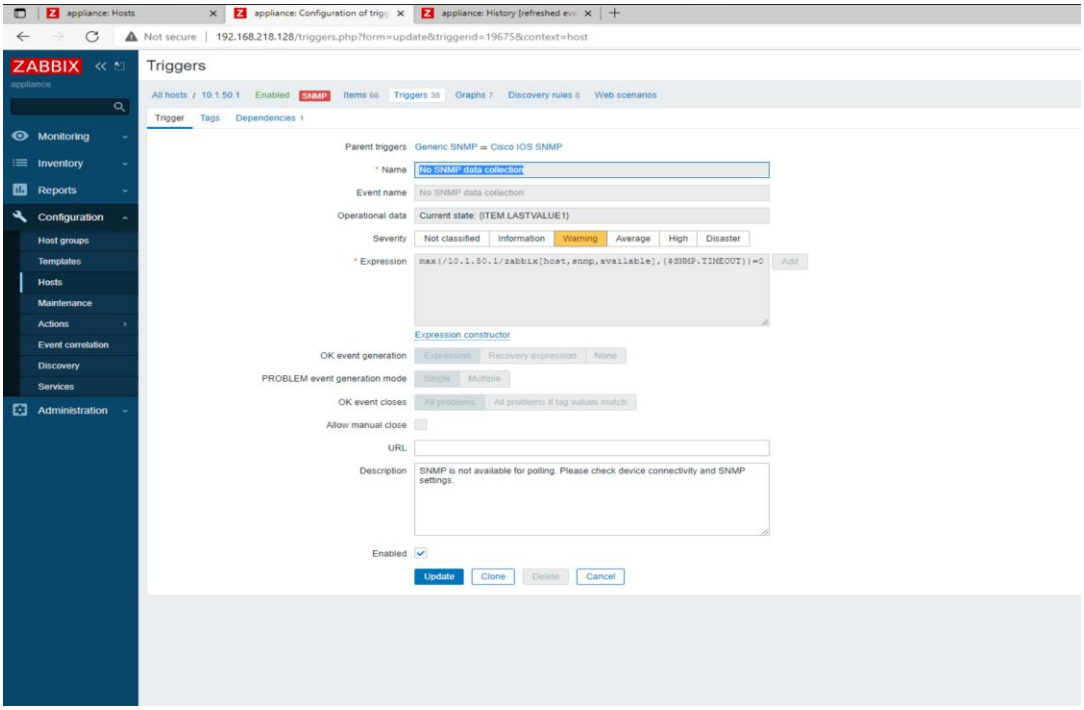


Figure 6.13: *Another critical checkpoint configuration* by S. Khan, 2022.

6.5.2 Implementation of Passive Performance Monitoring

The framework's objective is to analyse the run-time traffic dataset of the real-time network. Chapter 4.2 discusses the detail of the passive service monitoring framework. We develop a real-time network monitoring environment through a network simulation, as discussed in section 6.5.1. In this stage, we collect the traffic dataset from this run-time environment and perform four distinct steps to analyse the dataset. Therefore, the passive performance monitoring approach's implementation process comprises four steps, detailed in the following section.

Step 1: Capturing Network Traffic

We used Wireshark, the traffic capture software, to capture the network from a run-time network. A screenshot of the SDN traffic capture process is shown in Figure 6.14. This

captured traffic data is utilised for the next step of the passive network performance monitoring framework.

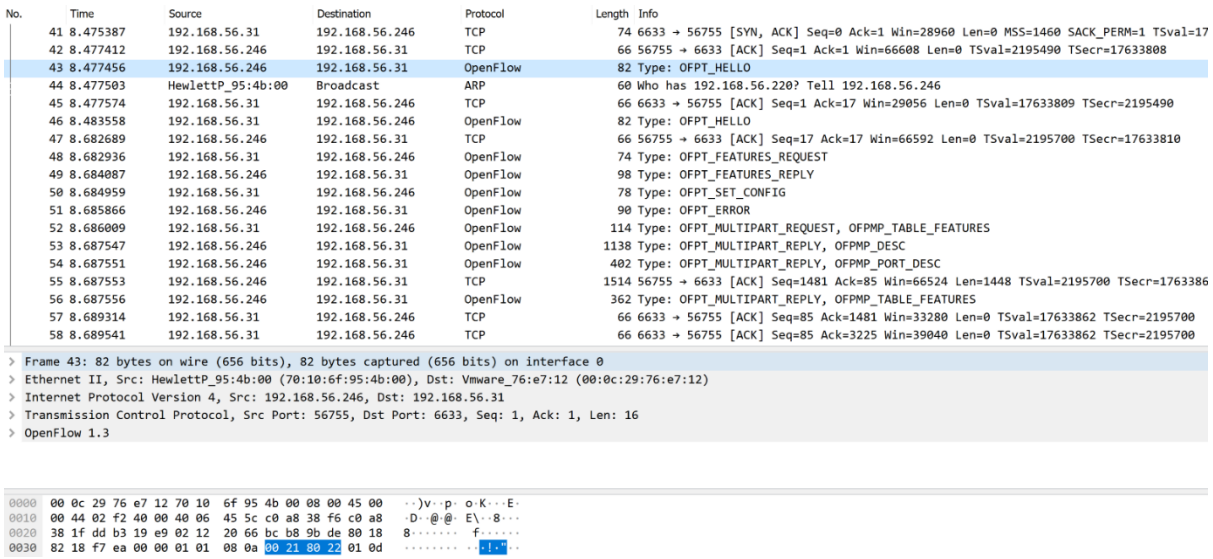


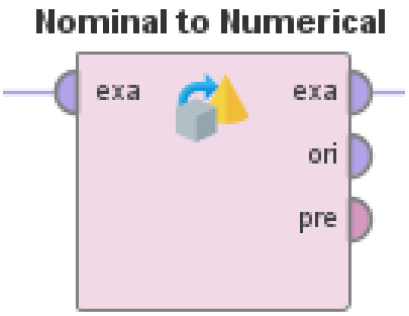
Figure 6.14: *A screenshot of the captured SDN traffic* by S. Khan, 2022.

Step 2: Identify Network Traffic Type

Identifying the network traffic type as an approach is discussed in Section 6.4.2. A machine learning-based algorithm (support vector machine) is used to determine the traffic type from an unknown set of traffic. The rapid Miner software is used to execute the machine learning technique. The process consists of the following stages.

Stage 1: We need a network traffic-labelled dataset to perform this task. Thus, the SDN traffic dataset is collected from the UNSW-NB15 dataset (*Cloud Stor*), and due to the size of the dataset, we used the first 300 traffic data and retrieved the traffic dataset into the rapid Miner using the Retrieve operator, as shown below.

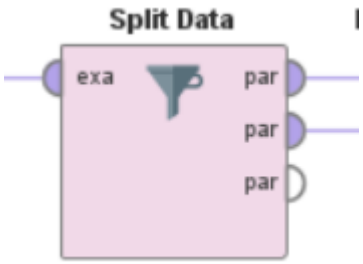
Step 2: The collected traffic dataset consists of nominal and numerical data. We converted the nominal data into numerical data using the "nominal to numerical" operator.



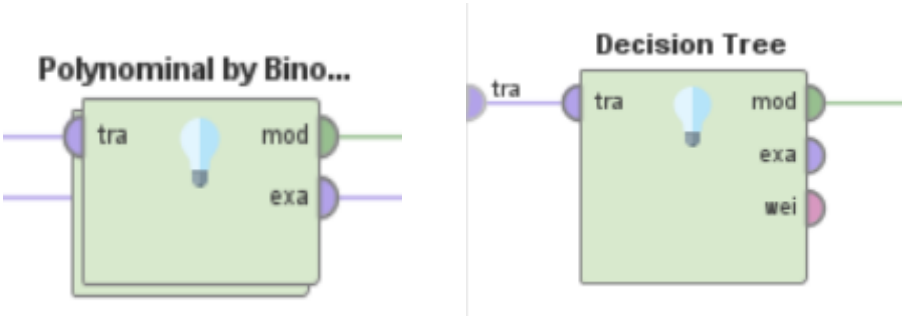
Step 3: To set the role for the model, we define the attribute name as category and the target role as Label.

Step 4: We split the dataset into two groups to train the model. The group of labelled data that will train the model is called the training dataset, and the target dataset from which we want to obtain the output is the testing dataset. In our experiment, we used the following combinations

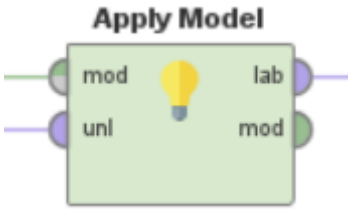
of training and testing datasets: Training dataset: Testing dataset = 80%: 20%, 90%: 10% and 60%: 40%.



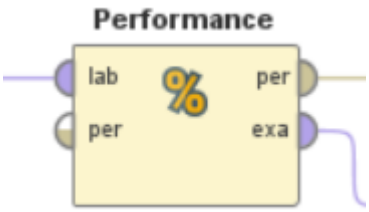
Step 5: We used the polynomial to a binominal operator to implement the decision tree model.



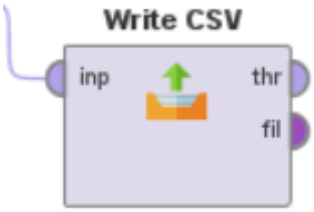
Step 6: The model operator is used to apply our trained model to our testing dataset and identify the category of the unknown traffic or testing traffic.



Step 7: The performance operator is used to measure the accuracy of the trained model.



Step 8: We use the Write CSV.; the operator assists in accessing the result in a CSV format for better understanding.



The implementation process screenshot is provided in Appendix A. In this stage, we used our network traffic data. We have some pre-selected labelled datasets where the network traffic has been categorised. We used the machine learning technique to identify the network traffic type. We trained our model using our labelled dataset, tested it with several variations, and achieved 96% accuracy. This means our model successfully identified the traffic and the accuracy rate is 96%.

Step 3: Classify Network Traffic

Network traffic classification is a process of categorising the traffic type based on business demand. The details of the approach are discussed in section 6.4.2. To achieve the objective of categorising the traffic data, we perform the following three steps:

Step 1: Step 2 discusses a machine learning-based approach to identifying traffic. We categorise the traffic into six groups based on the service type available in the collected dataset. The classification is primarily inspired by the Cisco Type of Service Classification approach. Based on our used dataset, we find 14 different types of traffic. On the other hand, Cisco

discussed eight traffic categories identified in their referenced book(Cisco Systems, 2014). Therefore, we have considered all the mentioned traffic types in this research.

Step 2: According to the identified traffic type of our dataset, we classified the entire traffic according to its service type. We first introduce six basic categories and align the 14 different types of traffic within these categories.

Step 3: As mentioned in section 6.4.2 (Step 3), we classified the six sub-categories into an advanced form and found three significant categories. The category details are discussed in section 6.4.2(Step 4).

Main Category	Sub-category	Traffic Category type from the dataset	Traffic category types from Cisco
Business relevant	**Network and Control Plane Protocol (Priority 1)**	Network	Network Control
		VPN	Signalling
		Cloud	Operations/ Administration/ Management (OAM)
	Voice Application (Priority 2)	VoIP	Voice
	Video Application (Priority 3)	Streaming	Multimedia Streaming
		Video	Multimedia Conferencing
			Real-time Interactive
			Broadcast Video
	Data Application (Priority 4)	Email	Transactional Data
		Collaboration	
		Web	
Default	**Default Forwarding (Priority 5)**	Software Update	Bulk Data
		Download-File Transfer-File Sharing	
		System	
Business Irrelevant	**Business Irrelevant Category (Priority 6)**	Music	
		Media	
		Social Network	
		shopping	

Table 6.5 details the traffic classification process and considers all the identified categories.

Three primary categories are developed, which include six sub-categories of traffic. Python programming language is used to define the traffic classification process. The results are detailed in the results section.

Step 4: Priorities Traffic

Prioritising network traffic includes valuing the traffic category based on dependency and severity. We have prioritised the traffic category using our proposed method, discussed in section 6.4.2 and demonstrated in Table 6.7. The method is inspired by the traffic prioritisation process developed by Cisco, shown in Table 6.6.

Number	Name
0	Routine
1	Priority
2	Immediate
3	Flash
4	Flash-override
5	Critical
6	Internet
7	Network

Table 6.6: IP Precedence Values by Cisco ISO Release 15 M&T(Cisco Systems, 2013)

Main Category	*Sub-category*	*Traffic Category type in the dataset*	*Additional traffic category type from Cisco*
Business relevant	**Control Plane Protocol (Priority 1)**	Network (1)	Network Control (1)
		VPN (3)	Signaling (1)
		Cloud (7)	Operations/ Administration/ Management (OAM) (1)
	Voice Application (Priority 2)	VoIP (4)	Voice (4)
	Video Application (Priority 3)	Streaming (5)	Multimedia Streaming (5)
		Video (5)	Multimedia Conferencing (5)
			Real-time Interactive (5)
			Broadcast Video (5)
	Data Application (Priority 4)	Email (6)	Transactional Data (6)
		Collaboration (5)	
		Web (2)	
Default	**Default Forwarding (Priority 5)**	Software Update (10)	Bulk Data (10)
		Download-File Transfer-File Sharing (9)	
		System (8)	
Business Irrelevant	**Business Irrelevant Category (Priority 6)**	Music (13)	
		Media (14)	
		Social Network (16)	
		Shopping (15)	

Table 6.7: Traffic category priority number

We implement the traffic prioritisation method using the Python programming language. To maintain consistency, we choose Python programming language as the most suitable programming language for our framework because we can use Python for SDN controller programming, and we are able to implement every module of the SLA-based QoS guarantee framework for SOA-based SDN. The traffic prioritisation implementation results are described in the results section.

6.5.3 Results and Discussion

This section explains the results of our proposed service management framework. As previously mentioned, our proposed service management framework consists of two approaches, and the approaches and their implementation details are described in earlier

sections. This section describes the two service performance monitoring approaches followed by the discussions. The results of these two approaches are detailed in the following section.

6.5.3.1 Results of Proactive Performance Monitoring

We describe the results of the proactive service performance monitoring in this section.

Results of Zabbix monitoring

Figures 6.15 and 6.16 show the network's monitoring status with customised features, which is the front end of the Zabbix monitoring tool. Zabbix allows various dashboard features to be configured based on user requirements. Thus, various monitoring features are applied; for example, we set up SNMP monitoring for every device and established SNMP communication among all connected devices. Figure 6.17 shows that SNMP is successfully enabled on all devices added to the Zabbix monitoring dashboard.

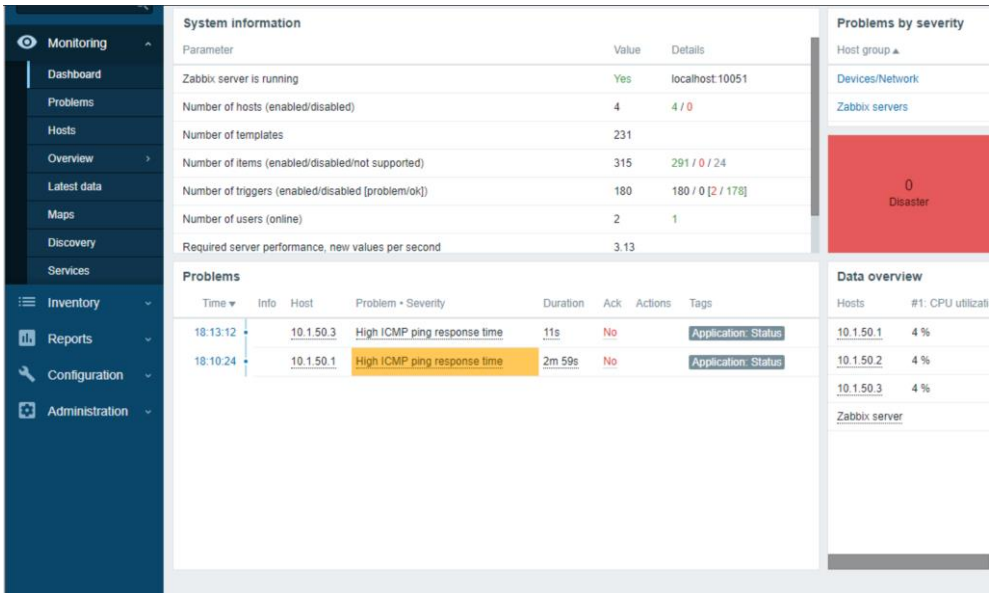


Figure 6.15: *Zabbix monitoring dash board with system information* by S. Khan, 2022.

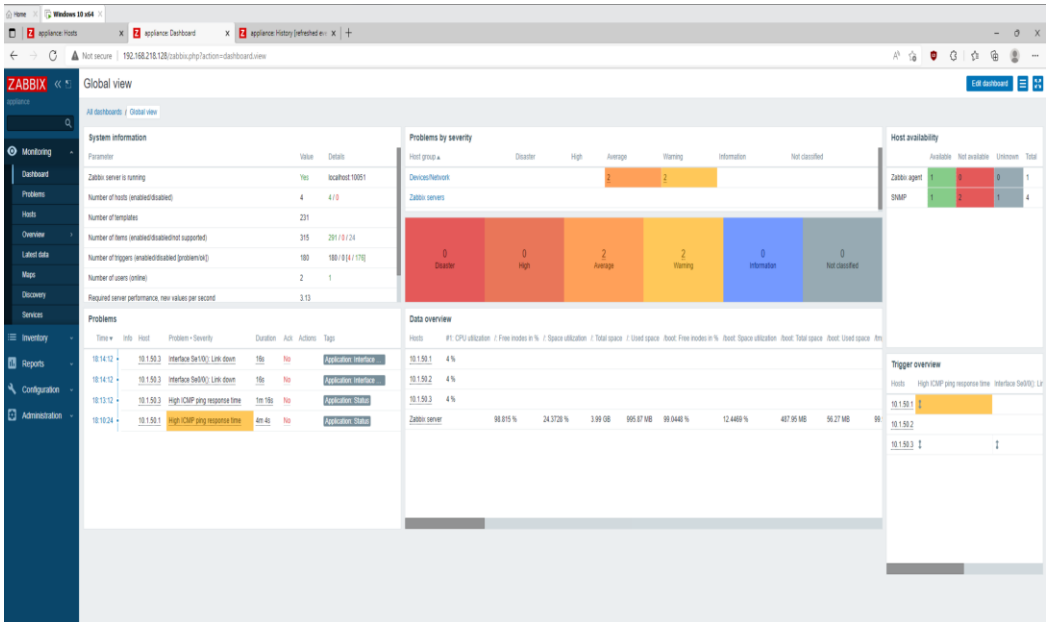


Figure 6.16: *Zabbix monitoring dashboard* by S. Khan, 2022.

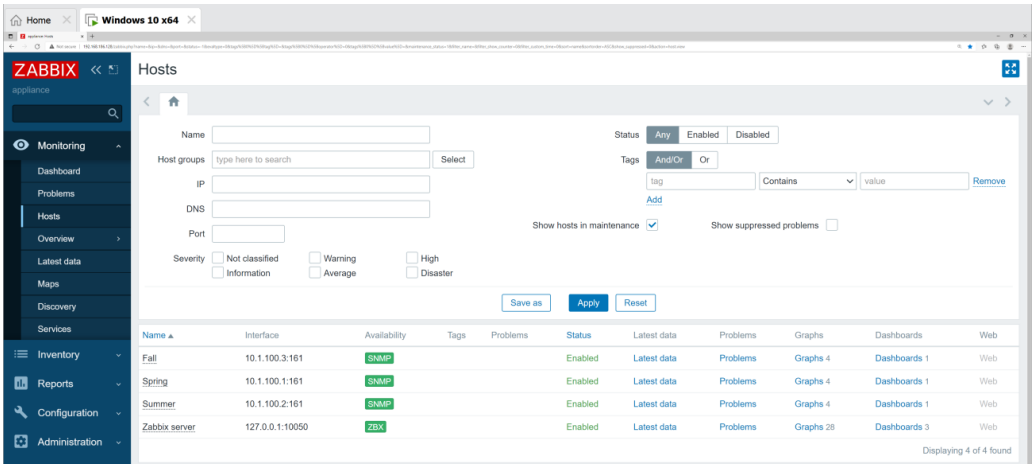


Figure 6.17: *SNMP monitoring is added to the network devices in the Zabbix monitoring dashboard* by S. Khan, 2022.

A global view of the Zabbix monitoring dashboard is illustrated in Figure 6.18, which demonstrates that four hosts are currently enabled and communicating with Zabbix. 315 items are currently available and operating to perform the monitoring task; of these them, 291 items are using SNMP communication. In the same way, 180 triggers, by default, are operating.

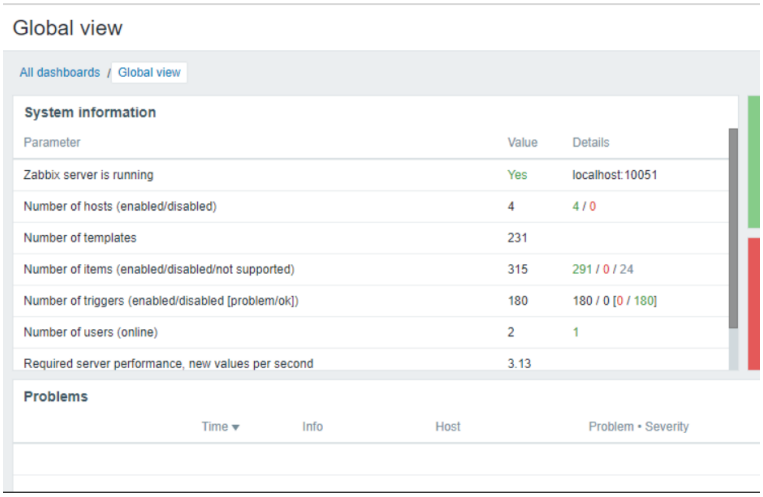


Figure 6.18: *Global view of the Zabbix monitoring dashboard* by S. Khan, 2022.

The data overview of Figure 6.19 demonstrates the status of the hardware and networking devices currently connected and monitored by Zabbix. The data overview shows various important information about the hardware devices, such as accessible, utilised, and used memory, ICMP loss, ping, and response time.

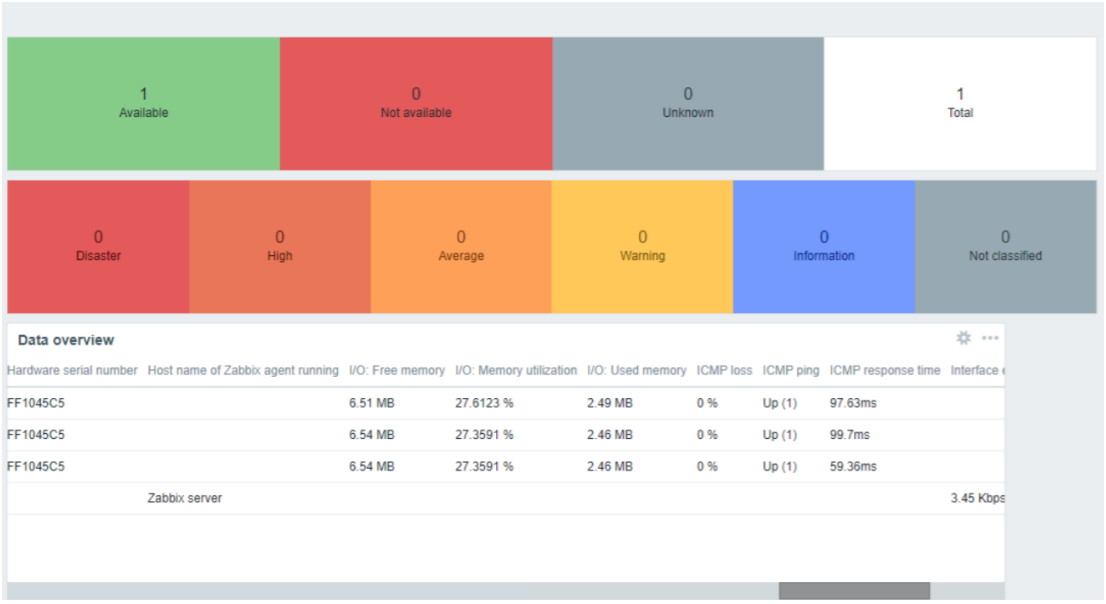


Figure 6.19: *Data overview of the Zabbix monitoring dashboard* by S. Khan, 2022.

Furthermore, the Zabbix dashboard allows us to see the network traffic using various graphs. Zabbix has the ability to collect traffic and graph information based on a predefined time interval. Figure 6.20 shows the time interval-based network traffic of the connection fast ethernet fa2/0 and fast ethernet 3/0 of the device with the IP 10.1.50.1. Moreover, Figures 6.21, 6.22, 6.23, and 6.24 illustrate the network traffic of the interfaces of the Fall router interfaces in a graph.

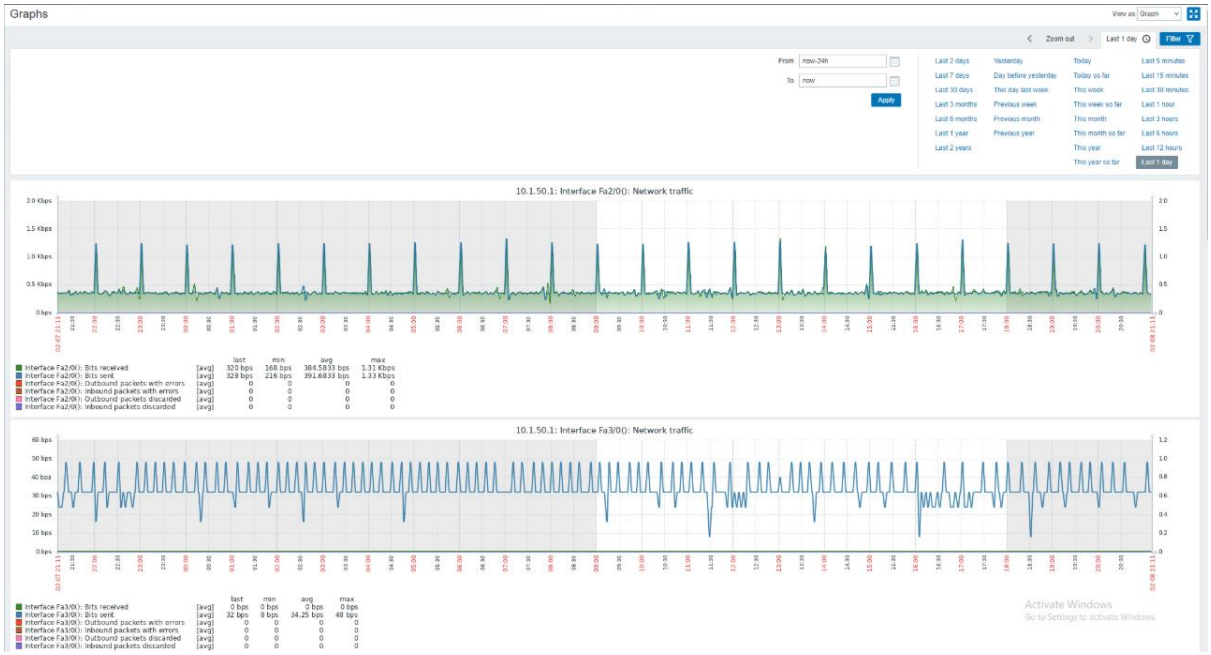


Figure 6.20: *Network traffic status showing in a graph of the spring device for the FastEthernet 2/0 with IP 10.1.50.1* by S. Khan, 2022.

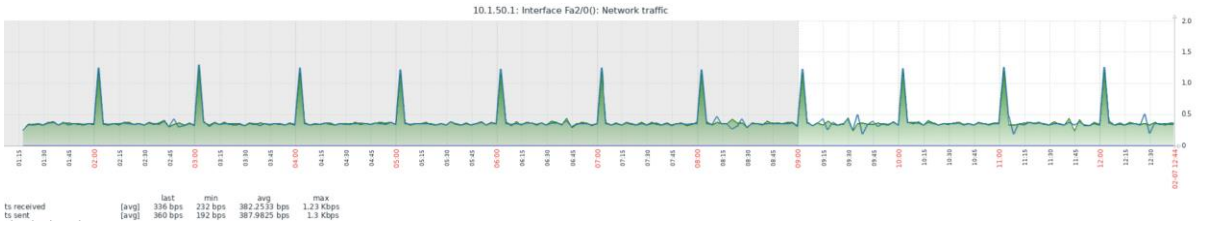


Figure 6.21: *Network traffic for the fast ethernet 2/0* by S. Khan, 2022.

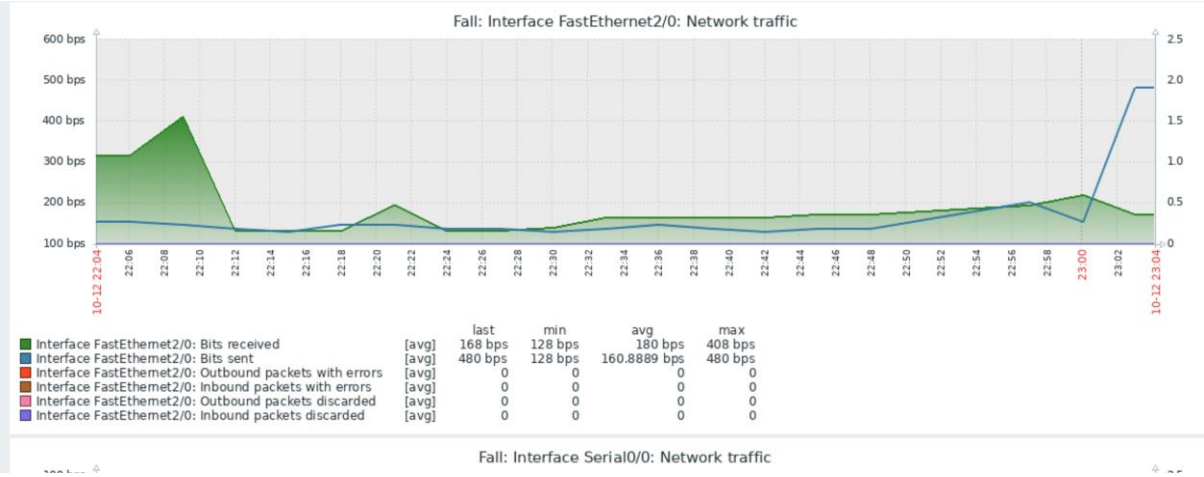


Figure 6.22: *Network traffic status of the fall router for the FastEthernet 2/0 with IP 10.1.50.1* by S. Khan, 2022.

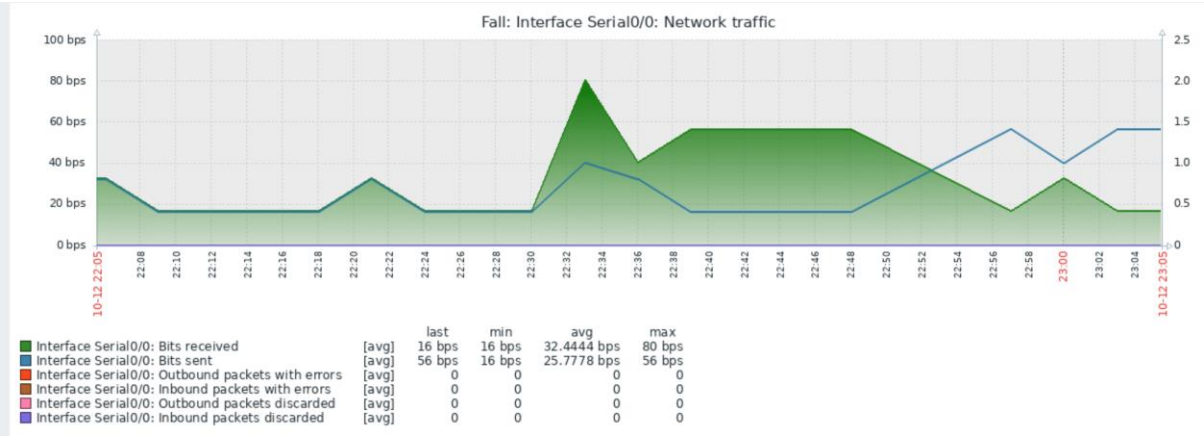


Figure 6.23: *Network traffic status is shown in a fall device graph for the serial interface 0/0.* by S. Khan, 2022.

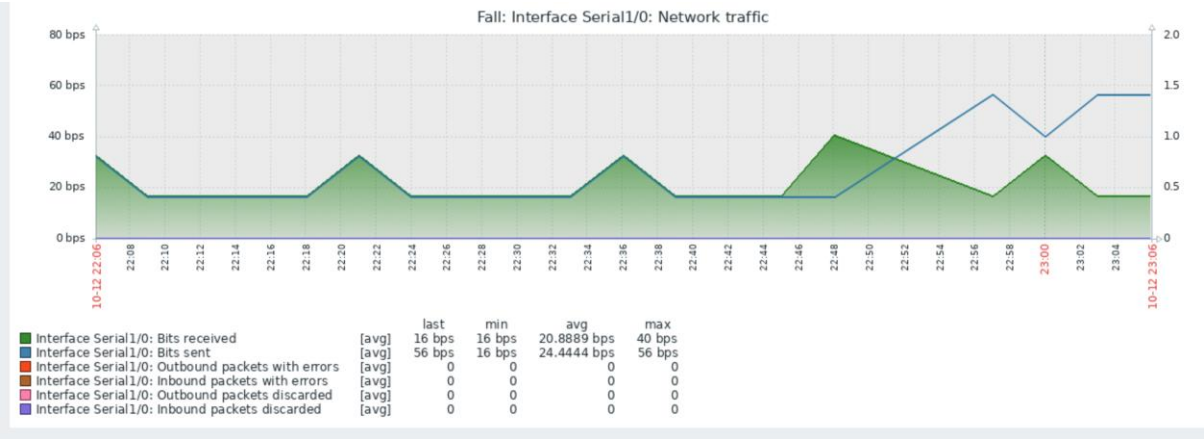


Figure 6.24: *Network traffic status showing in a graph of the fall device for the serial interface 1/0* by S. Khan, 2022.

Applying a trigger for the early checkpoint threshold and SLA threshold

We have successfully configured the item and trigger. We configured triggers for the critical checkpoint threshold and SLA checkpoint threshold. We tested our configured trigger, and the results show that the triggers are successfully activated, as shown in Figures 6.25 and 6.26. We can also observe that our configured monitoring agent is able to activate the alert according to our setup severity. A time-wise activated trigger is shown in Figure 6.27.

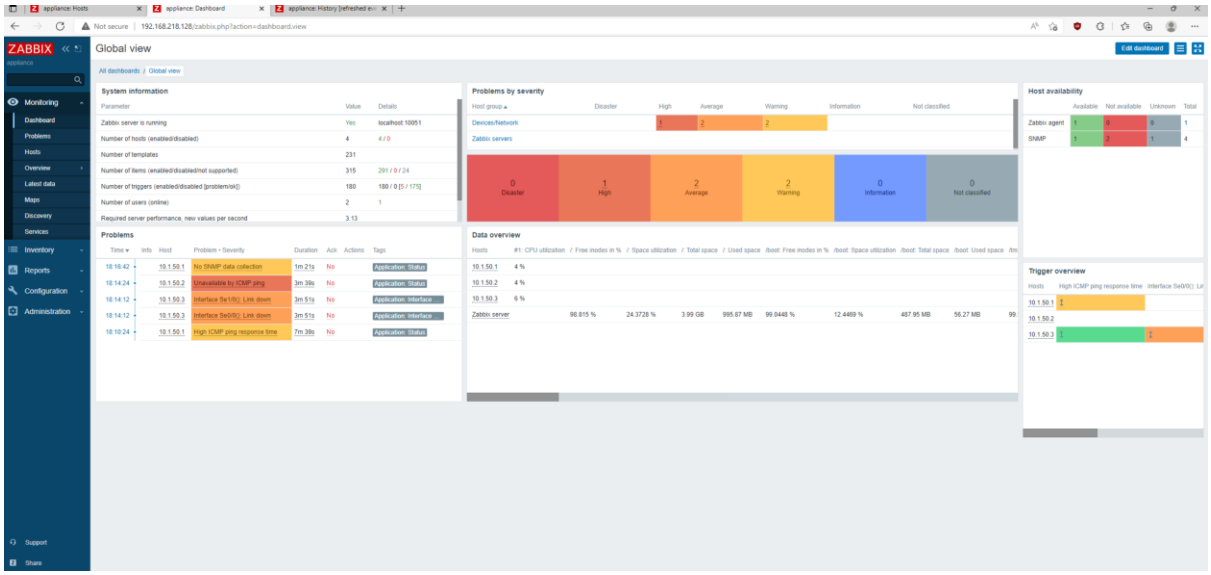
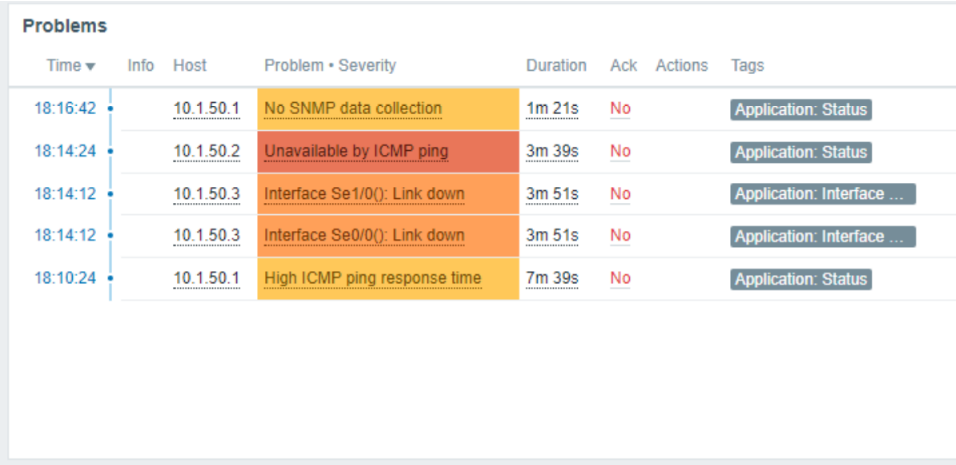


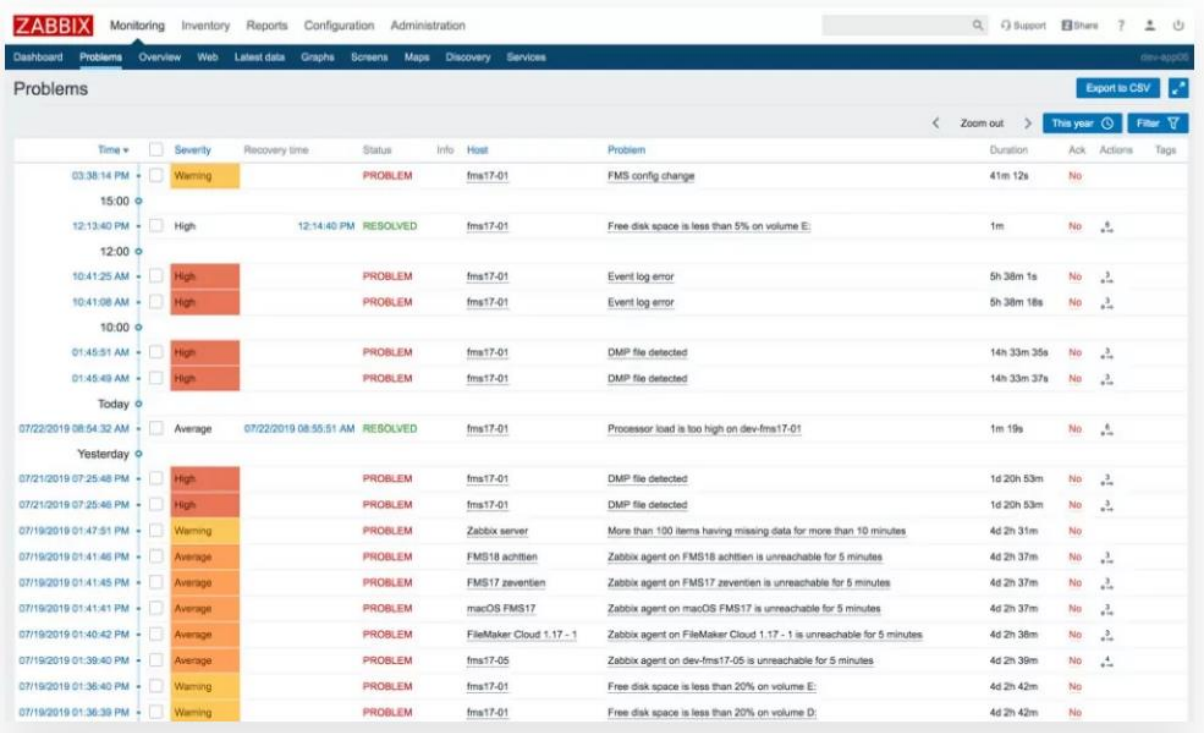
Figure 6.25: *Trigger activation* by S. Khan, 2022.



Problems

Time ▾	Info	Host	Problem • Severity	Duration	Ack	Actions	Tags
18:16:42		10.1.50.1	No SNMP data collection	1m 21s	No		Application: Status
18:14:24		10.1.50.2	Unavailable by ICMP ping	3m 39s	No		Application: Status
18:14:12		10.1.50.3	Interface Se1/0(): Link down	3m 51s	No		Application: Interface ...
18:14:12		10.1.50.3	Interface Se0/0(): Link down	3m 51s	No		Application: Interface ...
18:10:24		10.1.50.1	High ICMP ping response time	7m 39s	No		Application: Status

Figure 6.26*: Trigger activation with severity* by S. Khan, 2022.



ZABBIX Monitoring Inventory Reports Configuration Administration

Dashboard Problems Overview Web Latest data Graphs Screens Maps Discovery Services

Problems

Export to CSV

Zoom out This year Filter

Time ▾	Severity	Recovery time	Status	Info	Host	Problem	Duration	Ack	Actions	Tags
03:38:14 PM	Warning		PROBLEM		fms17-01	FMS config change	41m 12s	No		
15:00										
12:13:40 PM	High	12:14:40 PM	RESOLVED		fms17-01	Free disk space is less than 5% on volume E:	1m	No		
12:00										
10:41:25 AM	High		PROBLEM		fms17-01	Event log error	5h 38m 1s	No		
10:41:08 AM	High		PROBLEM		fms17-01	Event log error	5h 38m 18s	No		
10:00										
01:45:51 AM	High		PROBLEM		fms17-01	DMP file detected	14h 33m 35s	No		
01:45:49 AM	High		PROBLEM		fms17-01	DMP file detected	14h 33m 37s	No		
Today										
07/22/2019 08:54:32 AM	Average	07/22/2019 08:55:51 AM	RESOLVED		fms17-01	Processor load is too high on dev-fms17-01	1m 19s	No		
Yesterday										
07/21/2019 07:25:48 PM	High		PROBLEM		fms17-01	DMP file detected	1d 20h 53m	No		
07/21/2019 07:25:46 PM	High		PROBLEM		fms17-01	DMP file detected	1d 20h 53m	No		
07/19/2019 01:47:51 PM	Warning		PROBLEM		Zabbix server	More than 100 items having missing data for more than 10 minutes	4d 2h 31m	No		
07/19/2019 01:41:46 PM	Average		PROBLEM		FMS18 achttien	Zabbix agent on FMS18 achttien is unreachable for 5 minutes	4d 2h 37m	No		
07/19/2019 01:41:45 PM	Average		PROBLEM		FMS17 zeventien	Zabbix agent on FMS17 zeventien is unreachable for 5 minutes	4d 2h 37m	No		
07/19/2019 01:41:41 PM	Average		PROBLEM		macOS FMS17	Zabbix agent on macOS FMS17 is unreachable for 5 minutes	4d 2h 37m	No		
07/19/2019 01:40:42 PM	Average		PROBLEM		FileMaker Cloud 1.17 - 1	Zabbix agent on FileMaker Cloud 1.17 - 1 is unreachable for 5 minutes	4d 2h 38m	No		
07/19/2019 01:39:40 PM	Average		PROBLEM		fms17-05	Zabbix agent on dev-fms17-05 is unreachable for 5 minutes	4d 2h 39m	No		
07/19/2019 01:36:40 PM	Warning		PROBLEM		fms17-01	Free disk space is less than 20% on volume E:	4d 2h 42m	No		
07/19/2019 01:36:39 PM	Warning		PROBLEM		fms17-01	Free disk space is less than 20% on volume D:	4d 2h 42m	No		

Figure 6.27: *A detailed view of all the triggers activated with severity and time* by S. Khan, 2022.

6.5.3.2 Results of Passive Performance Monitoring

The results of the passive performance monitoring approach are described in the following section.

Results of identifying traffic type using the machine-learning-based approach

The objective of the implementation is to identify the traffic successfully. The machine learning-based implementation approach is discussed in the implementation section. Table 6.8 shows a portion of the result where the prediction _category depicts the category of the specific flow identified using our developed machine-learning-based support vector machine (SVM) model. The original _category depicts the origin of the specific traffic flow category. The results show that our developed model is able to identify or predict traffic flow successfully with 96% accuracy, and the results are shown in figure 6.28 below.

flow_key	src_port	dst_port	prediction_category	original_category
b26d66636dec141d796754edb97a1937	53409	443	Video	Video
b1e158d68b3f1ffe850d5f4a807473ec	60145	53	Web	Web
e6381671cc7f2b1714a4f9612b6386e0	65009	80	Web	Web
1307abc01bc662f2e9f568811b0e2e37	59422	993	Web	Web
d6b4b37ea173a995848381ad9c92c6b0	16403	16385	Web	Web
bab50edfa8e5c2b16fd6ddf53effdc31	59216	443	Web	Web
a44ab4fae95ee8ba7078f2420a60c524	63572	53	Web	Web
c4260a2a8e3c9e8c90f956625e4f0e3e	50795	3128	Streaming	Streaming

d36ffcd797c4c9e8911f38e039dcfbc6	59215	443	Streaming	Streaming
a93d4eb2f3cf54b3fc4b418eafbb9654	51400	53	Cloud	Cloud
f79a35e4b13b1593dab6503fed32d9b1	49433	53	Software Update	Software Update
f85ed541fe96ec5367b98892bbc0527c	32903	443	VPN	VPN
6ebdf34bf547e58bfbdb706cb30ff5f9	59013	80	Web	Web
047e627434a7c6f908d66b140e92f8ae	59687	443	Web	Web
f70a89bbe367607aa347b02a4587ae24	42296	443	Web	Web
2e74bbdefad0a184760b976c6c7632f9	49904	443	Web	Web
86de994eb8141a277910ac3ea4008d36	55558	53	Network	Network
134731c0a619602927ed2a9f8ba3bf10	51938	53	Cloud	Cloud
4e0995963f9f4ef0d77365b41024f2ad	63597	17000	Music	Download-File Transfer-Filesharing
db42092f9644cf3f4e53666932bd27ed	52621	443	Social Network	Social Network
ac65e1b542853233dcb06520cb845d1b	59463	443	Collaborative	Collaborative
b450fe7da015ced3900f639deb172be9	54248	443	Collaborative	Collaborative
5bcd4ab222f43ace41b3f7eb647cde53	58707	3128	Email	Email
a25fbe268237168d12fd80837f2140a7	50666	465	Email	Email
478f4ffb74e231f94a90ebc5b4383329	57586	53	Web	Web
16bb0783dc4e0029052dfd30b8867e78	52879	53	Web	Web
5040abace65d4b6805640c50aad1b8d6	53326	995	Web	Web

Table 6.8: A snapshot of the machine learning-based traffic identification result

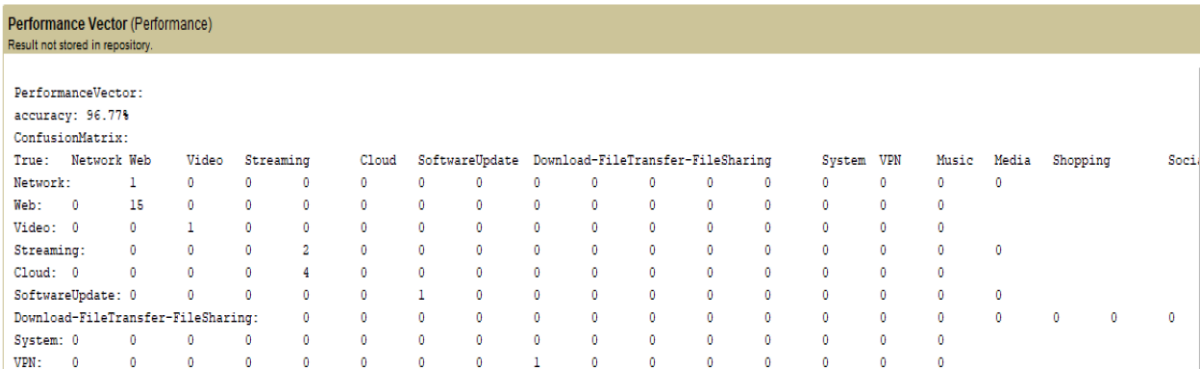


Figure 6.28: Traffic identification accuracy results using the Support Vector Machine(SVM) approach.

Results of traffic prioritisation

Table 6.9 details the results of traffic prioritisation. As mentioned in the implementation section, we classified the traffic into three main groups and six subgroups. Subsequently, we prioritise the group according to our proposed traffic prioritisation approach.

We employed Python programming to program the traffic classification and prioritisation rule. Then, we imported the dataset into the program and determined the priority values of the dataset. We identified the six primary priority values (priority) and 14 sub-priority values (priority_val) detailed in Table 6.9. The results show that the traffic classification and the prioritisation program can classify the network traffic and automatically prioritise them successfully.

flow_key	src_port	dst_port	prediction_category	original_category	priority	priority_val
b26d66636dec141d796754edb97a1937	53409	443	Video	Video	3	11
b1e158d68b3f1ffe850d5f4a807473ec	60145	53	Web	Web	1	2
e6381671cc7f2b1714a4f9612b6386e0	65009	80	Web	Web	1	2
1307abc01bc662f2e9f568811b0e2e37	59422	993	Web	Web	1	2
d6b4b37ea173a995848381ad9c92c6b0	16403	16385	Web	Web	1	2
bab50edfa8e5c2b16fd6ddf53effdc31	59216	443	Web	Web	1	2
a44ab4fae95ee8ba7078f2420a60c524	63572	53	Web	Web	1	2
c4260a2a8e3c9e8c90f956625e4f0e3e	50795	3128	Streaming	Streaming	3	12
d36ffcd797c4c9e8911f38e039dcfbc6	59215	443	Streaming	Streaming	3	12
a93d4eb2f3cf54b3fc4b418eafbb9654	51400	53	Cloud	Cloud	1	7
f79a35e4b13b1593dab6503fed32d9b1	49433	53	Software Update	Software Update	2	10
f85ed541fe96ec5367b98892bbc0527c	32903	443	VPN	VPN	1	3
6ebdf34bf547e58bfbdb706cb30ff5f9	59013	80	Web	Web	1	2

047e627434a7c6f908d66b140e92f8ae	59687	443	Web	Web	1	2
f70a89bbe367607aa347b02a4587ae24	42296	443	Web	Web	1	2
2e74bbdefad0a184760b976c6c7632f9	49904	443	Web	Web	1	2
86de994eb8141a277910ac3ea4008d36	55558	53	Network	Network	1	1
134731c0a619602927ed2a9f8ba3bf10	51938	53	Cloud	Cloud	1	7
4e0995963f9f4ef0d77365b41024f2ad	63597	17000	Music	Download-File Transfer-file Sharing	2	9
db42092f9644cf3f4e53666932bd27ed	52621	443	Social Network	Social Network	3	16
ac65e1b542853233dcb06520cb845d1b	59463	443	Collaborative	Collaborative	1	5
b450fe7da015ced3900f639deb172be9	54248	443	Collaborative	Collaborative	1	5
5bcd4ab222f43ace41b3f7eb647cde53	58707	3128	Email	Email	1	6
a25fbe268237168d12fd80837f2140a7	50666	465	Email	Email	1	6

Table 6.9: A sample of the result of traffic prioritisation.

6.5.3.3 Discussion

The experiment results show that our proposed service management framework consisting of proactive and passive service performance monitoring frameworks can monitor the current performance of the SOA-based SDN. Moreover, the framework assists in the run-time monitoring of the network and takes appropriate action proactively if the service performance is degrading. In addition, passive performance monitoring facilitates an analysis of the current performance of the service deliverables; therefore, the service monitoring agent defines the rules in the run-time network. The proposed framework assists the service provider in taking appropriate action to preserve the SLA before a violation occurs. Therefore, the framework can fulfil objective 2, which is to develop an intelligent framework to monitor the SDN services and provide predictive decisions of pro-active service violations. Furthermore, in the discussion section, we use the following viewpoint to argue that it makes the framework more functional, reliable, and efficient enough to serve the purpose.

1. Implementing the critical checkpoint and SLA checkpoint are primarily the triggers of the Zabbix monitoring tool. This trigger can be customised and applied according to the SLA and business demand. Therefore, the options are always open to introducing

various performance parameters or additional parameters in the framework and employing the checkpoints to monitor the service performance.

2. The discussion of the proposed framework and the experiment are evidence that the framework can successfully monitor the service performance from a run-time system. Moreover, the framework can analyse the current service performance and, according to the results, the approach prioritises their services and is able to fine-tune the run-time monitoring rules to prevent any possibility of a violation of the SLA contract and assist in preserving the SLA. Thus, the framework prevents service providers and consumers from facing penalties and additional aggravation.

3. A machine learning-based solution is proposed to identify the traffic type. The support vector machine model is successfully employed and receives the results with 96% accuracy. The proposed approach is unique compared to the existing approaches, such as port-based discovery, traffic length-based discovery, etc., which are used to identify traffic types.

4. The proposed framework prioritises services based on SLA and business needs. Therefore, the framework can be applied in any business dimension, such as small, medium, and large, not for any fixed type of business.

5. The system we developed can perform with a dynamic range of values according to business and SLA demand. This option opens up the opportunity to use the developed system in SDNs and various types of other networks, such as SOA-based military tactical networks and heterogeneous radio networks.

The above discussion shows the success of introducing various additional features that enable the customisation of the proposed service management framework to monitor the SOA-based SDN-oriented architectures successfully and preserve the trust relationship between the service consumer and provider.

6.6 Conclusion

In this chapter, we present a framework using proactive continuous service performance monitoring and passive service performance monitoring to ensure the service is delivered within the range in terms of quality that is expressed in the SLA. Therefore, this chapter makes a critical contribution to managing the SLA. The framework is named the Service Management Module. Proactive continuous performance monitoring is designed to proactively monitor the performance of the service consumer and the service provider against the service performance in the SLA. Moreover, we proposed a passive service performance framework that is designed to analyse the current traffic, including identifying the network traffic type, determining the vulnerability in terms of ensuring QoS, and employing the traffic priority in terms of service demand to achieve the committed QoS.

The primary objective of the service management framework is to continuously monitor the service-based SDN on a run-time basis and employ some intelligent rules to preserve the SLA. Instead of applying one-off monitoring, we proposed a predefined time window-based continuous performance monitoring method to avoid service failures. Instead, our time window session-based monitoring assists the third-party agent in being proactively involved in the network management activity to preserve the SLA that also significantly impacts the basis of the success of trust maintenance. We demonstrate proof of concept-based implementation of the framework in Section 6.5. The implementation results and the discussion of the proposed framework in section 6.5.3 demonstrate that the proposed framework assists the service provider in taking appropriate action to preserve the SLA before a violation occurs. Therefore, the developed service management framework is able to achieve objective 2, namely, to develop an intelligent framework to monitor the SDN services and provide predictive decisions of pro-active service violations. Furthermore, in the discussion, we argue from the following viewpoint that it makes the framework more functional, reliable, and efficient enough to serve the purpose.

6.7 References

Braun, R. (2016). Someone know about the a data set for software defined networks? Someone know about the a data set for software defined networks?

Chang, E., Hussain, F., & Dillon, T. (2006). *Trust and reputation for service-oriented environments: technologies for building business intelligence and consumer confidence*. John Wiley & Sons.

Cisco Systems, I. (2013). *qos : classification Configuration Guide, Cisco IOS Release 15 M&T*.

Cisco Systems, I. (2014). *Cisco IOS Quality of Service Solutions Configuration Guide* (Vol. 18). American Headquarters, Cisco Systems, Inc.

Cisco Systems, I. (2017). *Cisco EasyQoS Solution Design Guide* (1.6 ed.). Cisco.

Cloud Stor.

Fachrunnisa, O. (2011). *A methodology for maintaining trust in virtual environments*, Curtin University].

Keller, A., & Ludwig, H. (2003). The WSLA framework: Specifying and monitoring service level agreements for web services. *Journal of Network and Systems Management, 11*(1), 57-81.

Quillinan, T. B., Clark, K. P., Warnier, M., Brazier, F. M., & Rana, O. (2010). Negotiation and monitoring of service level agreements. In *Grids and Service-Oriented Architectures for Service Level Agreements* (pp. 167-176). Springer.

Stephanie Overby, L. G., Lauren Gibbons Paul. (2017). What is an SLA? Best practices for service-level agreements. https://www.cio.com/article/2438284/outsourcing-sla-definitions-and-solutions.html

Chapter 7

SERVICE EVALUATION AND VIOLATION PREDICTION FRAMEWORK FOR IMPARTIAL AND TRUSTFUL SERVICE DELIVERY IN SOA-BASED SDN

7.1 Introduction

Trust relationship preservation and service reliability share a harmonious relationship due to their explicit dependency on each other. Service reliability is the probability that a certain service will perform its intended function adequately for a specific period. In other words, service reliability expresses the ability to operate in a defined environment to promise service dependably and accurately without failure. The trust component is collected, codified, analysed, and evaluated evidence correlating to completeness, honesty, security, or dependability to make assessments and decisions regarding trust relationships (Staab et al., 2004). Therefore, we can conclude that trust relationship preservation can be obtained if the required level of reliability or confidence can be achieved in the deliverable service.

The trust component comprises various elements, such as identifying and collecting data or information, codifying the system, and analysing, assessing, or evaluating the framework to build confidence that the system is operating in accordance with its commitment. Therefore, we need a process or approach that can evaluate the framework and ensure that the proposed framework is operating with completeness, honestly, and securely and assists in decision-making by performing an impartial assessment. This framework will assist in achieving reliability or confidence in the deliverable services and preserves the trust relationship.

Chapter 7 presents a service evaluation and violation prediction framework for impartial and trustful service delivery in SOA-based software-defined networking (SDN). The framework is implemented during the interaction by a third party or an intermediate agent on which the interacting parties have agreed. Thus, the proposed framework facilitates the maintenance of trust using service performance records. This is needed to ensure that both parties deliver services as closely as possible to the service level agreement (SLA) that is agreeing to the mutual agreement by the end of the interaction. Our framework is anticipated to achieve reliability and trust preservation in service-oriented SDN environments. The fundamental

hypothesis of our SLA-based framework is that a trust relationship has already been established during interaction and requires sustaining a trust relationship by consistently monitoring, examining, and evaluating the interaction. Hence, both parties need to have a transparent picture of the deliverable services by evaluating the deliverable services and identifying any SLA violation during the interaction. As we know, any violation involves penalties, including financial and reputation. Hence, both parties need a framework that detects a violation and can predict any possible violation. This chapter presents our service evaluation and violation predicting framework for preserving SLAs and achieving service reliability.

Section 7.2 describes the details of the framework. Section 7.3 explains the framework implementation, results, and discussion, and section 7.4 concludes the chapter.

7.2 Proposed Service Evaluation and Violation Prediction Framework for Impartial and Trustful Service Delivery in SOA-based SDN

This section presents our proposed service evaluation and violation detection framework to provide impartial and trustful service delivery in SDN. The proposed framework assists both the service consumer and the service provider in ascertaining the truthfulness of the interaction and the reliability of the deliverable services. Moreover, the framework also impartially evaluates the execution of the SLA and preserves the SLA for future continuation.

After ascertaining the QoS monitoring log that is extracted from the passive performance monitoring approach of the service (Chapter 6), it is important to evaluate the QoS according to predefined criteria in the SLA and discover if any actual violation of the contract has occurred before deciding to continue with the SLA. For the service consumer, the proposed framework gives a broader understanding of the status of the deliverable services in terms of the contract based on QoS, and for the service provider, it provides an accurate picture of the QoS of the deliverables in terms of contract violation. Considering both the service consumer's and the service provider's perspectives enables both parties to assess each other on different

aspects before deciding to continue the trusting relationship between them. As shown in Figure 7.1, the proposed framework consists of two stages as follows:

stage *1: Formalisation of the quantifiable services*

In this stage, the intermediate or neutral evaluation agent interacts with the service consumer and service provider, collecting all related information regarding the deliverable services and articulating this information according to the predefined measurable metrics for the service evaluation process. The details of the formalisation of the quantifiable services stage are described in the next section.

stage 2: Condition evaluation service

In this stage, the intermediate or neutral evaluation agent performs the evaluation process regarding the condition of the agreement or the QoS parameter metrics committed to in the SLA. The evaluation agent receives information on the current QoS status and formalisation the quantifiable service information from the previous stage. This stage analyses and identifies any service discrepancies that may occur in the case of service-level contract violation. The details of the condition evaluation service are described in the next section.

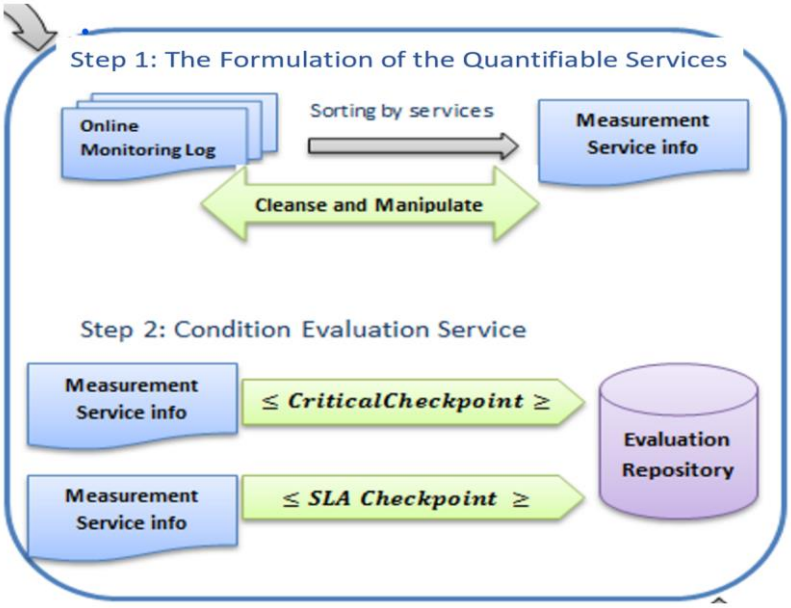


Figure 7.1: *Service evaluation and violation prediction framework* by S. Khan, 2022.

.

7.2.1 The Formalisation of Quantifiable Services

The formalisation of quantifiable services is an approach that we use to develop the service evaluation and violation detection and prediction framework. The approach can be compared to the measurement service process (Keller & Ludwig, 2003) due to the similarity of the approach used. The measurement service is a module that enables the measurement of all or a subset of SLA parameters (Keller & Ludwig, 2003). The measurement service contains information according to the current system configuration. It accesses runtime information by retrieving a resource matrix or from an external domain, such as the service provider's domain. In this research, we follow the measurement service process due to its unique features, such as a multiple measurement services capability that can simultaneously measure different matrices (Keller & Ludwig, 2003).

The formalisation of the quantifiable service approaches assists in preparing the raw dataset into quantifiable data according to the evaluation metrics. An intermediate entity or an agent, chosen by both parties (the service receiver and provider party), is actively involved in the framework; therefore, the service consumer and provider play no prominent role in the framework. As shown in Figure 7.2, the proposed approach comprises the following three steps:

Step 1: Collecting a Network Traffic Time-series Dataset

Collecting network traffic time-series datasets is one of the primary activities of the proposed service evaluation and violation detection framework. Since the network traffic time-series dataset is the critical element in operating the framework, we view this activity (network traffic time series dataset collections) as the foundation of the evaluation framework. We can portray this step as the input of the framework. Besides, collecting the network traffic time-series dataset consists of various quite complex steps, and the process details are explained in the previous service management framework (Chapter 6).

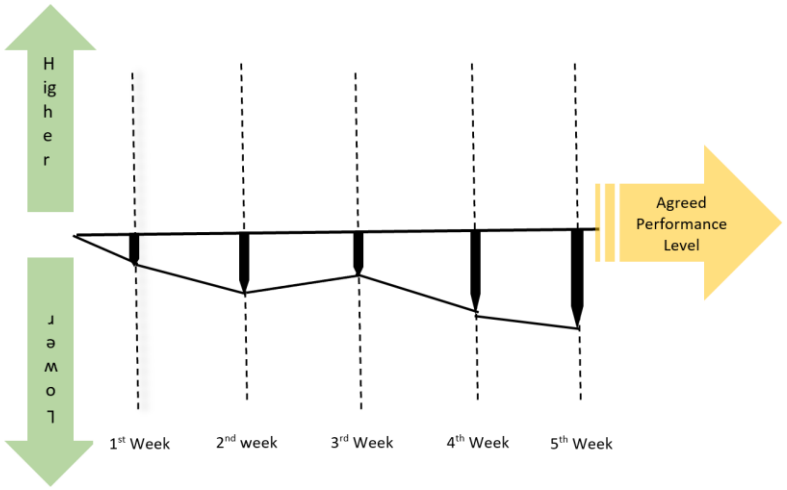


Figure 7.2: One-off performance evaluation reflects service unavailability over the five-week duration. Adapted from *One-off performance evaluation*, by Fachrunnisa, 2011.

As we want to collect the network traffic time-series dataset to evaluate the deliverable services, one-off traffic collection is not identified as ideal for this framework, as demonstrated in Figure 7.2. To ensure we do not overlook unseen concerns, we propose collecting a network traffic time-series dataset at various times during the overall interaction and considering the predefined time window described in the previous chapter (chapter 6, section 6.4.2.1). According to the proposed framework, the network traffic time-series dataset collection follows the predefined time-space where the interaction starts, and finish times are predefined for a specific time slot. Hence, we use the network traffic's time-series dataset from various arbitrary time windows to collect the network traffic dataset.

Step 2: Cleansing the Dataset

In this step, we cleanse our collective network traffic time-series dataset by identifying data errors and modifying the datasets by changing, updating, or removing data to correct them. The process is applied to fix the incomplete, duplicate, or incorrect data in the dataset. Thus, cleansing the dataset improves the data quality and assists in retrieving more consistent, accurate, and reliable information for decision-making. We cleaned our network traffic time-series dataset as closely as possible and prepared the datasets according to the evaluation metrics, and the process details are discussed in the next stage.

Step 3: Prepare the Dataset According to the Evaluation Metrics

In this step, we modify and prepare the dataset accordingly with the intention of unitising the prepared dataset for the next stage of the framework. We consider four performance parameters for the service evaluation and violation detection framework research to evaluate the network service performance. The definition of the four performance parameters and how to use them to measure performance are described below.

Performance parameter 1: Bandwidth

Bandwidth is the maximum amount of data or volume of information transmitted over a network channel in a specific amount of time. The units of bandwidth are measured in megabits per second (Mbps)(Cisco Systems 2005, 2006).

The process of calculating bandwidth:

- The total number of packets sent and received over a specific time.

Performance parameter 2: Throughput

Network throughput is defined as the amount of data transmitted successfully from one place or node to another place or node in a given time period. Throughput is typically measured in bits per second(bps), megabits per second (Mbps), or gigabits per second (Gbps)(Cisco Systems 2005)

The approach to calculate throughput:

- The quantity of data being sent/received by a unit of time
- The average rate of successful data transfer or
- Size of the transfer/ Time

Performance parameter 3: Jitter

A jitter is a change in the time it takes for a data packet to travel across a network and is usually caused by network congestion. On the sending side, packets are transmitted in a continuous

stream, and the packets are evenly spaced. However, due to network congestion, improper queuing or configuration errors, the steady stream can become lumpy, or the delay between each packet can vary instead of remaining consistent (Cisco Systems 2006). The performance of VoIP and video services may be negatively impacted due to the longer delay/jitter; hence VoIP jitter can make calls incoherent.

The approach to calculate Jitter:

As mentioned above, determining Jitter is measured by the time required to transmit a packet across a network(Cisco Systems 2005). Figure 7.3 demonstrate the time required to travel a packet from source to destination using the simple ping command.

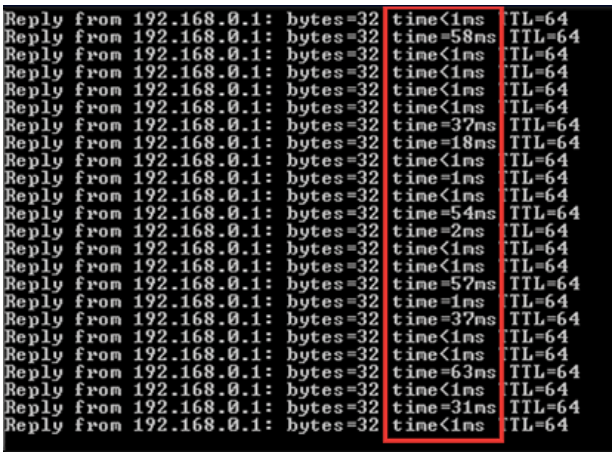


Figure 7.3: Ping command reply in a network.

Using the above figure, we discuss a simple process of determining Jitter in the following. The ping commands demonstrate the time difference between sending and receiving, and the differences are as follows:

- The time difference between 58 and 1 ms: 57 ms.
- The time difference between 1 and 58 ms: 57 ms.
- The time difference between 1 and 1ms: 0ms.
- The time difference between 37 and 1ms: 36 ms
- And similarly, continue the calculation.

There were 16 values (excluding seven 0ms) with an average difference of 41. It is 660 when it comes to the absolute difference. Then we take 660, divide it with the value 16, and receive nearly 41. Therefore, the Jitter of transmitting a packet from the source to the destination is 41 based on figure 7.3.

Performance parameter 4: Packet loss

Packet loss is the loss of packets of data not reaching their destination after being transmitted across a network. The common cause of dropped packets or packet loss is network congestion, hardware issues, software bugs, and several other factors.

Calculating packet loss: Calculate the number of packets lost per 100 packets sent by a host can be calculated using the following approach(Cisco Systems 2005):

- Network efficiency metric: Efficiency = 100% * (transferred - retransmitted) / transferred
- Network Loss = 100 – Efficiency

  Or
- Packet Loss Rate (PLR) = Total number of the packet transmitted – received packet/ transmitted packet * 100%

At the end of this stage, the network traffic time-series dataset is ready for deployment into stage 2 (the condition evaluation service approach) of the framework, and details are in the following section.

7.2.2 The Condition Evaluation Service

The condition evaluation service is responsible for comparing the current network performance against a predefined standard performance and receiving a result. The result will show whether a service discrepancy occurred in any timeslot of the interaction duration. The condition evaluation service approach comprises the following three steps.

Step 1 Employ the Service Degradation Critical Checkpoint Threshold and SLA Threshold

To employ the condition evaluation service, we need to define some standard performance ranges that the service can compare. Thus, we introduce acceptable performance thresholds in the framework at this stage. Figure 7.4 illustrates the purpose and actions of the threshold.

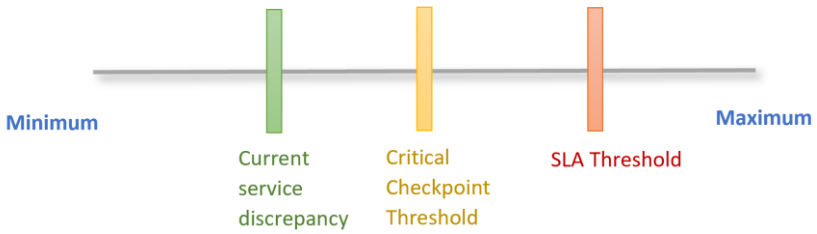


Figure 7.4: *Service degradation identification timeline* by S. Khan, 2022.

Let us assume that an SLA threshold is predetermined and defined in the SLA, which describes the maximum level of service degradation that is tolerable. If the service degradation exceeds the SLA threshold level, the services are considered to no longer align with expectations, and an SLA violation is deemed to have occurred, which results in the imposition of various penalties. Hence, to preserve the SLA, we introduce another threshold, the critical checkpoint threshold, which is implemented before the SLA threshold to pre-alert the interested parties that the service is degrading. For example, if the degradation of the SLA threshold is defined as 25, then the service degradation of the critical checkpoint threshold should be 20. The service performance degradation should be consistently lower than the critical checkpoint threshold to ensure the service performance is in a safer position and to preserve the SLA.

As previously discussed, implementing a critical checkpoint threshold plays a role in safeguarding the SLA threshold and allows us to predict the possibilities of SLA violation and intervene early to avoid a violation of the SLA. We apply the checkpoint and SLA threshold in

our framework. The implementation details and results of the approach are discussed in the experiments and results section.

Step 2: Evaluate the Service Performance with the Critical Checkpoint and SLA Threshold.

In this step, we evaluate the network traffic time-series dataset that gives an accurate picture of the current service performance of the SDN with the defined checkpoint threshold to ascertain how far the service degradation level is from the predefined checkpoint threshold level. Similarly, we evaluate the service performance against the SLA threshold to ascertain and assess the risk to the current service performance. Figure 7.5 shows the overall process of evaluation of the current performance with the critical checkpoint and SLA threshold.

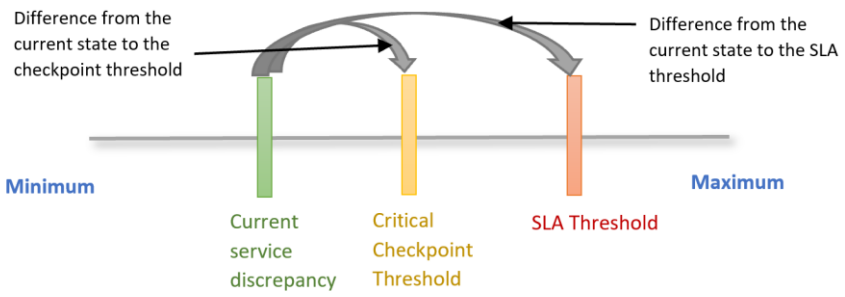


Figure 7.5: *Evaluating the traffic status data using the critical checkpoint and SLA threshold* by S. Khan, 2022.

Let us assume that the services running in the system show the performance is degrading. The current performance represents a measurement unit of 19. It is essential to evaluate the performance degradation with the critical checkpoint, and the difference is identified as 20. In this case, the third-party agent notifies the service provider by giving them scope for early intervention and taking appropriate action to reverse the performance until it reaches a safer position. Similarly, in other situations, if the service performance level is measured as 14 and after evaluating the current service degradation using the critical checkpoint threshold and SLA

threshold (15), it is evident to all interacting parties that the currently running service performance is in a critical stage and requires prompt action in order to preserve the SLA.

The experiments and results section describes the implementation details of this step, followed by a discussion of the results.

Step 3: Applying the Performance Discrepancy Condition and Labelled the Traffic

After evaluating the service performance, we employ some conditions that will assist us in ascertaining the overall current service performance status. Moreover, this step assists in processing the decision-making of the framework. Based on the previous discussion, one of four following scenarios may occur.

Scenario 1: current service performance is lower than the critical checkpoint threshold.

Scenario 2: current service performance is higher than or equal to the critical checkpoint threshold.

Scenario 3: current service performance is lower than the critical checkpoint but higher than the SLA checkpoint.

Scenario 4: current service performance is lower than the SLA checkpoint.

We translate the above four scenarios into four viable conditions that we employ in the framework. The conditions are as follows.

Condition 1: Current QoS > Critical checkpoint

Condition 2: Current QoS degradation <= Critical checkpoint

Condition 3: Critical checkpoint > Current QoS > SLA checkpoint

Condition 4: Current QoS <= SLA checkpoint.

These four quantifiable conditions and the labelled traffic are needed to engage in predictive decision-making. The details of the decision-making process are described in the following

section. The implementation of the above conditions in the framework is discussed in the experiments and results section.

Condition 1: Current QoS > Critical checkpoint, then no need to label as the current status is safe to operate.

Condition 2: Current QoS <= Critical checkpoint, then label as N_check

Condition 3: Critical checkpoint > Current QoS >SLA checkpoint, then label as N_check

Condition 4: Current QoS <= SLA checkpoint, then label as N_sla

7.3 Experiments and Results

The previous section describes the service performance evaluation and violation predicting framework comprising two stages. Each stage is implemented progressively and not only integrates the approaches but also integrates them with our previously developed framework (Chapter 6) to validate the framework's success. We engineered the system using the following tools to implement the service performance evaluation and violation prediction framework.

1. VMware, for virtualising and isolating the system environment
2. GNS3 emulator to develop the network prototype
3. Zabbix Appliance network monitoring tool to monitor the network prototype and deliver live updates
4. Python Programming

As discussed in Chapter 6, we developed the Cisco network prototype using GNS3 because, in terms of QoS traffic, there is no difference between Cisco performance network traffic data and SDN performance traffic data(Braun, 2016). Therefore, to reduce the complexity of the implementation, we developed a Cisco network prototype and continued our experiments based on the dataset that the prototype produced.

In this section, we discuss the implementation process of the framework and discuss the results. As previously mentioned, the implementation process of the two stages is as follows:

7.3.1 Implementation of the Formalisation of the Quantifiable Services

We have discussed the formalisation of the quantifiable services approach in the framework section (section 7.3). The approach comprises three steps, hence the experiments associated with these steps are as follows:

Step 1: Collecting time-series traffic dataset: Real-life data collection from a prototype simulation is one of the primary tasks of the framework. We used our previously set up network prototype environment using VMware and GNS3 with the Zabbix appliance monitoring tool. As we needed to collect time-series data, we followed the following procedure:

- Data is recorded per second, meaning each network traffic data record is added every second.
- Data collection takes over 10 minutes, as the time space is defined as 10 minutes.
- A random time slot is used for the data collection, in this case, half an hour for one week.
- Convert the collected data into a .csv file for further processing.

A segment of the collected raw dataset is presented below.

12878	12876	10075	Zabbix server: Number of processed log values per second	60	0	0	0
12879	12877	10075	Zabbix server: Number of processed log values per second	60	0	0	0
12880	12878	10075	Zabbix server: Number of processed log values per second	60	0	0	0
12881	12879	10075	Zabbix server: Number of processed log values per second	60	0	0	0
12882	12880	10075	Zabbix server: Number of processed log values per second	60	0	0	0
12883	12881	10075	Zabbix server: Number of processed log values per second	60	0	0	0
12884	12882	10075	Zabbix server: Number of processed log values per second	60	0	0	0
12885	12883	10075	Zabbix server: Number of processed log values per second	60	0	0	0
12886	12884	10075	Zabbix server: Number of processed log values per second	60	0	0	0
12887	12885	10075	Zabbix server: Number of processed log values per second	60	0	0	0
12888	12886	10076	Zabbix server: Number of processed numeric (unsigned) values per second	4	0.266310284	0.275792436	0.287543728
12889	12887	10076	Zabbix server: Number of processed numeric (unsigned) values per second	56	0.265792787	0.382373876	0.419563069
12890	12888	10076	Zabbix server: Number of processed numeric (unsigned) values per second	60	0.386827316	0.546878381	1.327853375
12891	12889	10076	Zabbix server: Number of processed numeric (unsigned) values per second	60	0.929419218	1.423672513	2.014164613
12892	12890	10076	Zabbix server: Number of processed numeric (unsigned) values per second	60	0.932026894	1.429690544	2.014262946
12893	12891	10076	Zabbix server: Number of processed numeric (unsigned) values per second	60	1.047692476	1.436591457	1.996071227
12894	12892	10076	Zabbix server: Number of processed numeric (unsigned) values per second	60	1.048717019	1.436277584	1.996526003
12895	12893	10076	Zabbix server: Number of processed numeric (unsigned) values per second	60	1.047350661	1.43328545	1.99624284
12896	12894	10076	Zabbix server: Number of processed numeric (unsigned) values per second	60	1.047784454	1.436379268	2.047855467
12897	12895	10076	Zabbix server: Number of processed numeric (unsigned) values per second	60	0.915220503	1.432426026	2.031258425
12898	12896	10076	Zabbix server: Number of processed numeric (unsigned) values per second	60	0.931064243	1.416902013	2.012301021
12899	12897	10076	Zabbix server: Number of processed numeric (unsigned) values per second	60	0.91423664	1.431702906	1.995618191

Table 7.1: A section of the network traffic dataset by S. Khan, 2022.

Step 2: Cleanse the dataset: We cleanse the time-series traffic dataset in this step. The raw dataset consists of various information and associated values, which is difficult to understand. We used a data cleansing process and prepared the dataset in a more understandable form.

Step 3: Prepare the dataset according to the evaluation metrics: In this step, we identified and prepared the network traffic time-series dataset accordingly so that it can interpret the network's current performance. Hence, we measured the four performance metrics: bandwidth, throughput, jitter/delay, and packet losses described in section 7.3. Table 7.2 shows a small segment of the prepared dataset from the 1441 records.

	A	B	C	D	E	F
1	Time Series	bandwidth	throughput	jitter	packet_loss	delay
2	1/04/2022 0:00	39494	35054	80.6	3.3	222
3	1/04/2022 0:01	24989	16755	94.8	3.2	495
4	1/04/2022 0:02	15461	6820	38.4	5.3	316
5	1/04/2022 0:03	3228	1135	126.5	1	982
6	1/04/2022 0:04	4077	3651	123.8	1.6	755
7	1/04/2022 0:05	8669	6854	74.7	5.2	732
8	1/04/2022 0:06	16848	8612	24	4.4	48
9	1/04/2022 0:07	34679	13128	43.4	1.3	194
10	1/04/2022 0:08	7038	3278	95.4	2.8	916
11	1/04/2022 0:09	44569	22128	31.2	2.8	109
12	1/04/2022 0:10	16859	15756	94.1	5.6	333
13	1/04/2022 0:11	28468	15552	110.5	3.7	1008
14	1/04/2022 0:12	36401	22530	70.4	4	435
15	1/04/2022 0:13	46825	23813	97.6	4.5	360
16	1/04/2022 0:14	2444	2036	30.4	3.5	295
17	1/04/2022 0:15	4066	1283	10.9	1.2	106
18	1/04/2022 0:16	35388	13529	96.9	4.9	227
19	1/04/2022 0:17	4805	2400	109	1.9	592
20	1/04/2022 0:18	26332	14380	119.5	3.6	1092
21	1/04/2022 0:19	33519	28293	123	2.9	152
22	1/04/2022 0:20	38374	24018	66.9	3.4	592
23	1/04/2022 0:21	11173	7146	24.3	3.1	133
24	1/04/2022 0:22	41061	13607	91.9	0.1	475
25	1/04/2022 0:23	10875	8438	19.2	5	177
26	1/04/2022 0:24	2059	1978	109.1	1.2	891
27	1/04/2022 0:25	28222	18741	112.2	1.1	225
28	1/04/2022 0:26	48338	43425	87.7	0.9	239
29	1/04/2022 0:27	28644	23264	52.6	4.5	166
30	1/04/2022 0:28	26003	9427	27.8	3.6	43
31	1/04/2022 0:29	8928	8104	103.6	2.3	394
32	1/04/2022 0:30	33322	17937	2.1	5.5	17
33	1/04/2022 0:31	11780	11209	18.9	1.9	50
34	1/04/2022 0:32	42062	38991	103	4.8	884
35	1/04/2022 0:33	48948	43067	121.7	0.6	415
36	1/04/2022 0:34	12699	11712	122.5	1.3	292

Table 7.2: Data preparation according to the evaluation metrics by S. Khan, 2022.

Completing steps 1, 2 and 3 concludes the implementation of stage 1(the formalisation of the quantifiable services) of the framework, and table 7.2 depicts the process of transforming unrefined network traffic data into refined and knowledgeable information, which is deployed in stage 2 (the condition evaluation service) of the framework.

7.3.2 Implementation of the Condition Evaluation Service

The condition evaluation service is a stage or module of the proposed service evaluation and violation prediction framework that enables impartial and trustful service delivery. As a result, the condition evaluation service plays a primary role in preserving the SLA. The framework's objective is not only to detect SLA violations, and it assists in predicting the possibility of SLA violation in a service-oriented SDN environment. From the condition evaluation framework discussion in section 7.2.2, we anticipate that the framework comprises three-step activities; therefore, we implement stage 2 in three steps. We used the Python programming language to implement stage 2, and the implementation process details are as follows:

Step 1: Implementing the service degradation critical checkpoint threshold and SLA threshold

To implement the service degradation critical checkpoint threshold and SLA threshold, firstly, we use the traffic categories we identified and explained in Chapter 6. Table 7.3 details the traffic categories, sub-categories and network traffic types we use in the experiments and any additional traffic category type defined by Cisco and included in our research.

Main Category	***Sub-category***	***Traffic Category type in the dataset***	***Additional traffic category types may need to be added from Cisco***
Business relevant	**Control Plane Protocol (Priority 1)**	Network	Network Control
		VPN	Signalling
		Cloud	Operations/ Administration/ Management (OAM)
	Voice Application (Priority 2)	VoIP	Voice
	Video Application (Priority 3)	Streaming	Multimedia Streaming
		Video	Multimedia Conferencing
			Real-time Interactive
			Broadcast Video
	Data Application (Priority 4)	Email	Transactional Data
		Collaboration	Bulk Data
		Web	
Default	**Default Forwarding (Priority 5)**	Software Update	
		Download-File Transfer-File Sharing	
		System	

Business Irrelevant	**Business Irrelevant Category (Priority 6)**	Music	
		Media	
		Social Network	
		shopping	

Table 7.3: *Identified traffic categories discussed in Chapter 6* by S. Khan, 2022.

Secondly, we determine the four performance measures for each traffic category. We identified the maximum and minimum bandwidth, delay, Jitter, and packet loss in each traffic category. However, to develop Table 7.4, we have made some reasonable adjustments where applicable. Table 7.4 shows each performance measure's minimum and maximum values with our predefined default values for each category.

At this point, we apply the checkpoint threshold and SLA threshold using the Python programming language. In a realistic scenario, the intermediate agent applies the critical checkpoint threshold and the SLA threshold in the system according to the predefined SLA contract which was agreed upon by all interacting parties. For this scenario, we used the maximum value as the SLA threshold default value and the medium value as the critical checkpoint threshold default value.

Hence, the system that we developed has the opportunity to enter its SLA threshold value and critical checkpoint threshold value. However, if no values are entered into the system, the system automatically collects the default SLA and critical checkpoint threshold values. The results section details the outcomes of this step.

Step 2: Experiment to evaluate the service performance using the critical checkpoint threshold and SLA threshold

In this step, we evaluate the current service performance of the network with the critical checkpoint threshold and SLA threshold. We used the dataset we prepared according to the evaluation metrics presented in table 7.2 above, representing the network's current performance.

Main Category	**Sub-category**	**Priority**	Bandwidth	Delay	Jitter	Loss
Business Relevant	**Conrol Plane protocol**	1	10 Mbit/s to 10Gbit/s Def: 1Gbit/s	Should be < 10ms to 100ms Def: <20ms	Should be <1ms to 10ms Def: <10ms	0%
	Voice Application	2	21 to 320 kbit/s Def: 300kbit/s	Should be < 150ms Def: <100ms	Should be< 30ms Def: <30ms	Should be < 1%
	Video (Interactive)	3	10Mbit/s to 900Mbit/s Def: 500Mbit/s	Should be < 150ms Def:<100ms	Should be <30ms Def: <30ms	Should be < 1% Def:<1%
	(Streaming)	3		<4/5s	No precise requirements	<5%
	Data Application	4	2.4 kbit/s to 200 Mbit/s Def: 150Mbit/s	Should be <10 ms to 50ms Def: <30ms	Should be < 30ms Def: 30ms	Should be <1% to 5% Def: <2%
Default	**Default Forwarding**	5	1kbit/s to 1 Gbit/s Def: 500Mbit/s	Should be <5/6s Def: <1s	Should be <30ms to50ms Def: <40ms	Should be <3% to 6% Def: <3%
Business Irrelevant Category	**Business Irrelevant Category**	6	1 kbit/s to 1Mbit/s Def: 1Mbit/s	Should be < 10s to 20s Def: <15s	Should be < 50ms Def: < 40ms	Should be <10% Def: <5%

Table 7.4: *Minimum and maximum values of the defined performance measures* by S. Khan, 2022.

As the dataset is very large, we first ascertain the average performance measurement metrics of the current QoS dataset. Thus, we have the average bandwidth, average throughput, average Jitter, and average packet loss of the current network status.

After receiving the current performance measurements or the average QoS performance values, we compare the value with the current checkpoint threshold. Similarly, we compare the current QoS performance values with the SLA values. Table 7.4 represents the minimum and maximum values of the defined performance measures of the predefined QoS metrics. Using reasonable assumptions, we determined the above QoS measures (Table 7.4) according to the main traffic category and sub-category. Python programming is used to perform the network service quality evaluation step. The outcome of this step is provided in the results section.

Step 3: Implement the performance discrepancy condition and the traffic labelling

We formulate four quantifiable performance discrepancy conditions in the framework and the current network performance or QoS status labelling process. We implement these in the system using Python programming and report the outcomes. Implementing this step results in an overall understanding of the status of the QoS, and the outcome of the implementation is discussed in the results section.

7.3.3 Results and Discussion

This section discusses the results of the proposed service evaluation and violation prediction framework. The simulation is designed to achieve objective 3 and examine the efficacy of our service evaluation and violation prediction framework. The examination is conducted by comparing the current network service performance variations and the QoS level with the SLA expectations. Moreover, the examination analyses the efficacy of our SLA management framework in SDN for informed decision-making. In this section, firstly, we discuss the results of the service evaluation and violation prediction framework, followed by discussing the achievements of the framework.

7.3.3.1 Results

The outcome of the condition evaluation and violation prediction framework is detailed in the following:

Collecting checkpoint threshold data:

The outcome of the framework retrieves and collects a set of data that reach or exceed the checkpoint threshold. This data collection process is performed based on the priority level with human involvement.

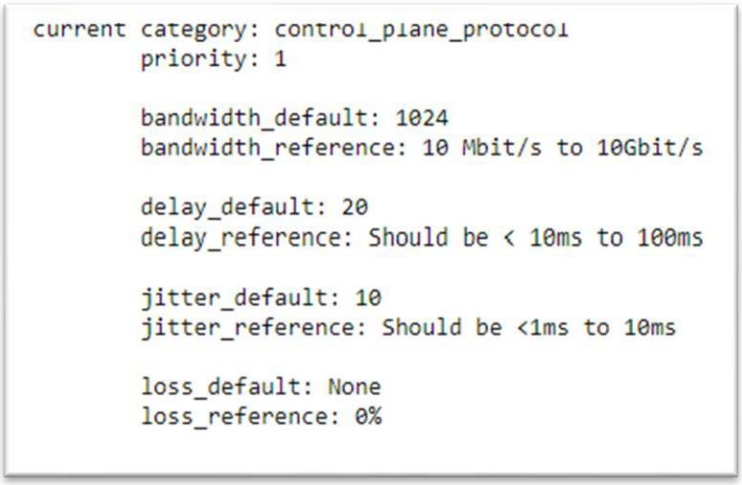


Figure 7.6: *Control_Plane_protocol (Priority 1) category traffic stored the default value* by S. Khan, 2022.

Figures 7.6 to figure 7.11 demonstrate the outcomes that our developed systems stored. The service evaluator or agent added their required values in the system according to SLA. The snapshots demonstrate the values that they have stored. If no values are entered, the system automatically takes the default two values we set up in the program to collect as the checkpoint threshold and SLA threshold values. The values are captured based on the priority level demonstrated in table 7.4.

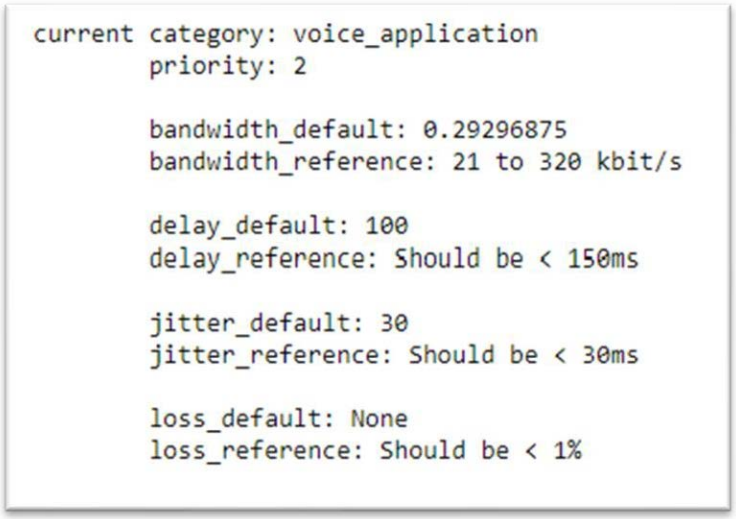


Figure 7.7: *Voice application category (Priority 2) traffic stored the default value* by S. Khan, 2022.

Figure 7.6 shows the default bandwidth, delay, Jitter and loss values of the control traffic category (priority 1) that is stored in the system as a checkpoint threshold. On the other hand, the reference bandwidth, reference delay, reference jitter and reference loss values are stored as the SLA checkpoint threshold value. The same process is continued for the voice application category (priority 2), video interactive category (priority 3), data application category (priority 4), default forwarding category (priority 5) and the irrelevant business category (priority 6) and the snapshots of these categories are demonstrated in figure 7.7, 7.8, 7.9, 7.10 and 7.11.

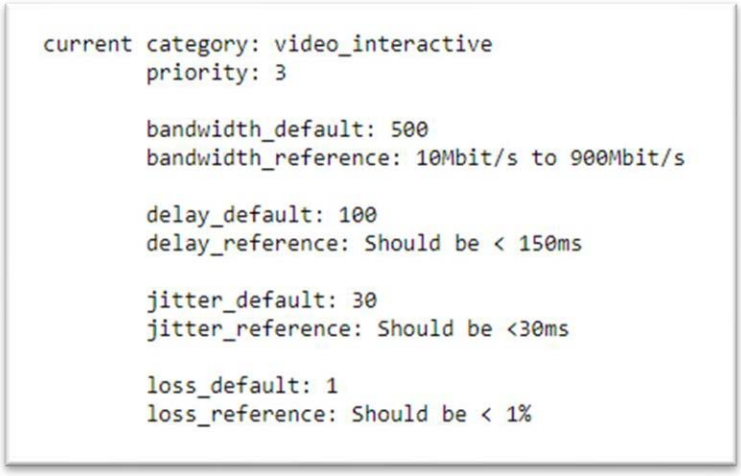


Figure 7.8: *Video_interactive (Priority 3) category traffic stored the default value* by S. Khan, 2022.

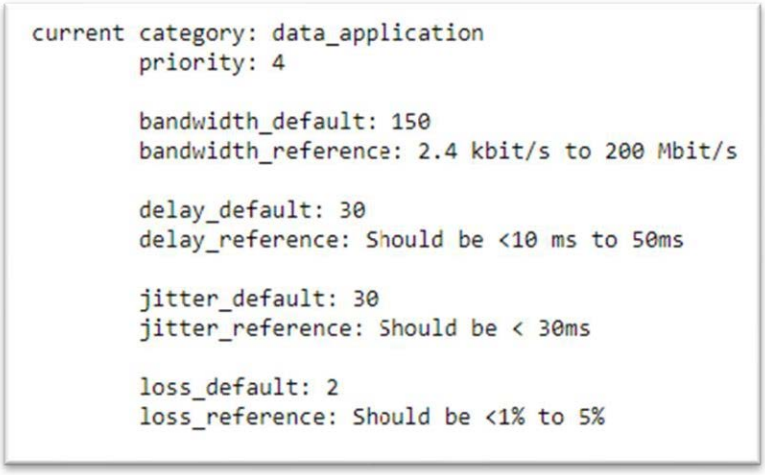


Figure 7.9: *Data_application category (Priority 4) traffic stored the default value* by S. Khan, 2022.

```
current category: default_forwarding
        priority: 5

        bandwidth_default: 4096
        bandwidth_reference: 1kbit/s to 1 Gbit/s

        delay_default: 1000
        delay_reference: Should be <5/6s

        jitter_default: 40
        jitter_reference: Should be <30ms to50ms

        loss_default: 3
        loss_reference: Should be <3% to 6%
```

Figure 7.10: *Default_forwarding category (Priority 5) traffic stored the default value* by S. Khan, 2022.

```
current category: business_irrelevant_category
        priority: 6

        bandwidth_default: 1
        bandwidth_reference: 1 kbit/s to 1Mbit/s

        delay_default: 15000
        delay_reference: Should be < 10s to 20s

        jitter_default: 40
        jitter_reference: Should be < 50ms

        loss_default: 5
        loss_reference: Should be <10%
```

Figure 7.11: *Business_irrelevant_category (Priority 6) traffic stored the default value* by S. Khan, 2022.

In addition, our developed systems collect the runtime network traffic dataset, determine the average QoS in terms of the QoS metrics we have defined and generate the results of our current network services QoS average values that are demonstrated in figure 7.12.

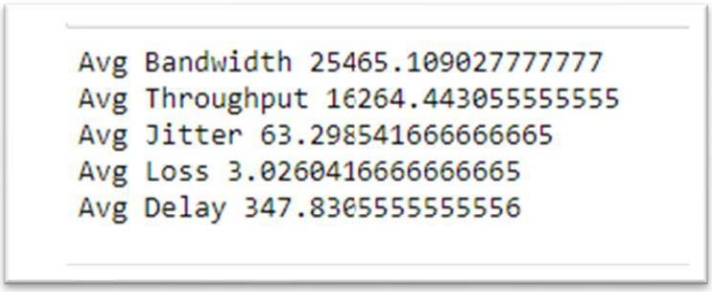


Figure 7.12: *Current network service QoS average* by S. Khan, 2022.

Figure 7.13 shows the critical checkpoint and SLA threshold values from the default entry. As mentioned in the discussions, the system we have developed can perform with a dynamic range of values according to business and SLA demand.

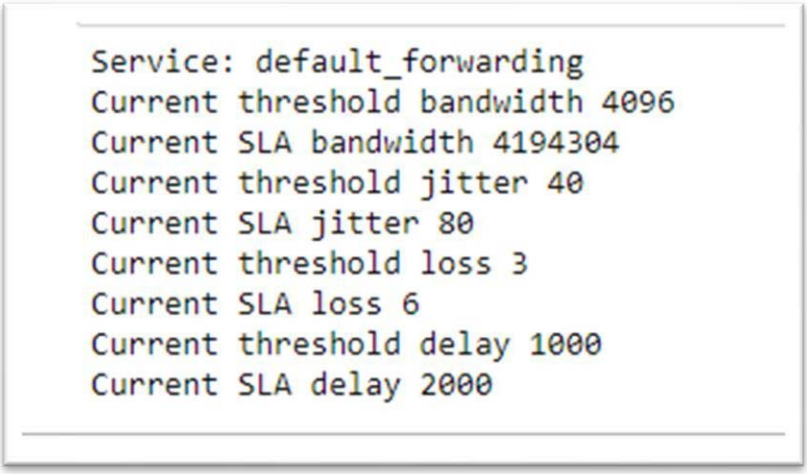


Figure 7.13: *Critical checkpoint threshold and SLA threshold stored in the system* by S. Khan, 2022.

ts	bandwidth	throughput	jitter	packet_loss	delay	n_sla	n_check
0	25638.333333	17136.833333	69.923333	3.160000	375.400000	False	True
1	27089.116667	17633.383333	62.513333	2.771667	319.133333	False	True
2	25026.766667	16066.566667	53.200000	2.998333	267.583333	False	True
3	24119.283333	15195.600000	58.033333	3.053333	338.666667	False	True
4	27373.466667	17731.766667	67.513333	2.835000	353.466667	False	True
5	24679.716667	15147.250000	63.151667	2.890000	371.183333	False	True
6	27110.633333	16469.016667	66.193333	2.800000	342.333333	False	True
7	25142.283333	15751.983333	58.951667	3.198333	315.600000	False	True
8	26781.550000	16573.400000	63.828333	3.178333	324.750000	False	True
9	25207.650000	16846.133333	65.590000	3.051667	343.950000	False	True
10	23900.233333	15297.900000	51.598333	3.391667	273.616667	False	True
11	21321.133333	13534.266667	73.776667	2.953333	443.450000	False	True
12	24912.100000	14888.750000	70.515000	3.415000	440.616667	False	True
13	23556.366667	16226.450000	67.408333	2.845000	406.683333	False	True
14	26144.150000	16476.650000	56.226667	3.326667	303.683333	False	True
15	26651.500000	16627.033333	69.346667	3.181667	380.766667	False	True
16	21821.200000	14201.016667	59.715000	3.156667	360.950000	False	True

Table 7.5: The labelled traffic

Table 7.5 shows the labelled traffic results collected through the evaluation process. The system compares the network performance's current status with the critical checkpoint threshold and labels the traffic as n_checks based on our threshold. Subsequently, the system similarly labelled the n_sla traffic. Table 7.5 above demonstrates the compared results, validates the predefined n_sla and n_check service prediction conditions, and shows the true and false results. The results show that though the performance of the current service has reached the critical checkpoint threshold quite a few times, no SLA violation has occurred.

From the above results, the service evaluation agent can realise the service's current status and predict that the service is at risk and requires close attention to reverse the service performance and ensure it is within an acceptable range.

7.3.3.2 Discussion

The experiment results show that the proposed service evaluation and violation prediction framework is able to evaluate the current performance of the SOA-based SDN. Moreover, the framework assists in predicting the possibility of a forthcoming violation of the SLA to help the service provider take appropriate action to preserve the SLA before the violation occurs. Furthermore, in the discussion, we argue that this makes the framework more functional, reliable, and efficient enough to serve the purpose.

1. The service evaluation and violation prediction framework experimentation considers four basic performance parameters to evaluate the network service performance. However, options are always open to introducing various performance parameters or additional parameters in the framework and evaluating the service performance.

2. The experiment results provide evidence that the framework can successfully evaluate the service performance from a runtime system. Moreover, the framework can detect SLA violations and predict the possibility of a violation of the SLA contract, assisting in preserving the SLA. Thus, the framework prevents both the service providers and consumers from facing unpredicted penalties that may result in significant financial loss and additional aggravation.

3. The proposed framework employs two checkpoints. However, we designed the framework considering the generic options for employing additional checkpoints according to the business strategies. Moreover, the framework is open to applying the critical checkpoint and SLA threshold values according to consumer expectations and the SLA contract.

4. The system we have developed can perform with a dynamic range of values according to business and SLA demand. This option opens up the opportunity to use the developed system not only in SDN but also in various types of other networks, such as SOA-based military tactical networks and heterogeneous radio networks.

The above discussion shows that introducing advanced features that enable the customisation of the proposed service evaluation and violation prediction framework enables the successful deployment of the framework in various SOA-based SDN architectures.

7.4 Conclusion

This chapter describes a service evaluation and violation prediction framework for impartial and trustful service delivery in SOA-based SDN. The framework is carried out during the interaction by a third party or an intermediate agent that the interacting parties agreed upon. This framework can evaluate and ensure that the services are operating with completeness, honestly, and securely and assists in decision-making by performing an impartial assessment. This framework assists in achieving reliability or confidence in the deliverable services and helps to preserve the trust relationship.

In this framework, both parties have runtime deliverable services data that can be used to evaluate service performance. The framework comprises two stages, namely formalisation of the quantifiable services and the condition evaluation service, to evaluate the performance of the deliverable services and predict the possibility of SLA violation. The implementation and validation results show that our proposed framework can neutrally evaluate service performance and predict the possibility of SLA violation which assists in preserving service reliability and the trust relationship. Therefore, we can conclude that the required level of trust relationship can be preserved if a required level of reliability or confidence can be achieved in the deliverable service.

7.5 References

Braun, R. (2016). Someone know about the a data set for software defined networks? Someone know about the a data set for software defined networks?

Cisco Systems (2005). Measuring Delay, Jitter, and Packet Loss with Cisco IOS SAA and RTTMON. https://www.cisco.com/c/en/us/support/docs/availability/high-availability/24121-saa.html#definedelayjitterpacketloss

Cisco Systems (2006). Understanding Jitter in Packet Voice Networks (Cisco IOS Platforms). https://www.cisco.com/c/en/us/support/docs/voice/voice-quality/18902-jitter-packet-voice.html

Fachrunnisa, O. (2011). *A methodology for maintaining trust in virtual environments*, Curtin University].

Keller, A., & Ludwig, H. (2003). The WSLA framework: Specifying and monitoring service level agreements for web services. *Journal of Network and Systems Management, 11*(1), 57-81.

Staab, S., Bhargava, B., Leszek, L., Rosenthal, A., Winslett, M., Sloman, M., Dillon, T. S., Chang, E., Hussain, F., & Nejdl, W. (2004). The pudding of trust. *IEEE Intelligent Systems, 19*(5), 74-88.

Chapter 8

SERVICE CONTINUITY DECISION MAKING

8.1 Introduction

Trust relationship preservation and service reliability share a harmonious relationship due to their explicit dependency on each other, as mentioned in the previous chapter (Chapter 7), leading to a significant service continuity association. Service continuity association can be defined as a particular unbiased decision-making interest in continuing the business interaction among the interacting parties, inspired by the probability of attaining a positive trust relationship by obtaining service reliability. On the other hand, service reliability is the probability that a particular service performs its intended function adequately for a specific period of time.

The previous chapter reports that there is a coherent relationship between service reliability and trust, where service reliability is considered to be the ability to operate in a defined environment to guarantee dependable and accurate service without failure. Likewise, the trust relationship implies there is a quantified belief in a trustor concerning a trustee's competence, honesty, security, and dependability within a specified context (Staab et al., 2004). Therefore, we can conclude that a positive trust relationship facilitates an impartial association of service continuity decisions if service reliability or confidence is achieved. Therefore, we require a framework that is able to give an impartial assessment of the deliverable services and ensure that the proposed framework operates with completeness, honesty, and security and assists decision-making. This framework should be a genuine representation of the outcome of the objective of delivering reliable services and preserving the trust relationship.

Chapter 8 presents an impartial and trustful service continuity decision-making framework for service delivery in service-oriented-architecture-based software-defined networking (SOA-based SDN). A third party implements the framework in the presence of the consumer during the interaction, especially at the midpoint of the interaction to which the interacting parties

agreed. As discussed in the previous chapter, the fundamental hypothesis of our SLA-based framework is that a trust relationship has already been established during an interaction and to sustain the trust relationship by consistently monitoring, examining, and evaluating the interaction. Hence, both parties need to have a transparent picture of the deliverable services by evaluating and identifying any SLA violation during the interaction. Moreover, we know that any violation incurs penalties, both financial and reputational. Hence, both parties need a framework that is able to give an impartial assessment of the interaction and assist in service continuation decision-making.

The chapter is organised as follows: Section 8.3 details the proposed service continuation decision-making framework. Section 8.4 discusses the implementation and results of the framework; and Section 8.5 concludes the chapter.

8.2 Proposed Service Continuity Framework

This section presents our proposed service continuity framework to assist impartial, informed service continuity decision-making in SOA-based SDN. The proposed framework assists the service consumer in analysing and understanding the overall performance of the SLA contract and deliverable services during the service interaction. This performance understanding helps the service consumer to make constructive and informed decisions as to whether they should continue the business relationship with a particular service provider or make alternative arrangements. On the other hand, the service provider ascertains the truthfulness of the interaction and the deliverable services resulting from that interaction. The impartial service performance information helps the service provider to identify room for improvement for future business transactions. Moreover, an impartial evaluation of the SLA execution using the framework discussed in Chapter 7 and our framework proposed in Chapter 8 ascertains whether the primary aim of preserving the SLA and the trust relationship is achieved.

After ascertaining the service performance evaluation results from the service performance evaluation and violation prediction framework, the current status of the services in terms of our

predefined checkpoint threshold, the SLA threshold, and service evaluation rules is determined. The service performance evaluation and violation prediction framework aims to evaluate the QoS according to predefined criteria in the SLA and discover any actual violation of the contract before deciding on continuing the SLA. We refer to our proposed service continuation framework as the continuation or advancement of our previous service performance evaluation and violation prediction framework.

In terms of the service consumer's point of view, the previous framework enables a broader understanding of the status of the deliverable services in terms of the agreed QoS. Moreover, our proposed service continuity framework assists in decision-making regarding continuing the business interaction with a particular provider based on an informed and impartial contribution.

Our previously developed service management framework assists the service provider in obtaining an accurate picture of the QoS deliverables in terms of predicting the possibility of an SLA violation and having the scope to take appropriate action before a violation occurs. In addition, using the proposed service continuation framework, only the consumer is able to analyse the service performance history and make the service continuation decision. On the other hand, the service provider does not have any involvement in the service consumer's decision-making; however, the service provider is able to understand the rationale for any decision made by a certain consumer.

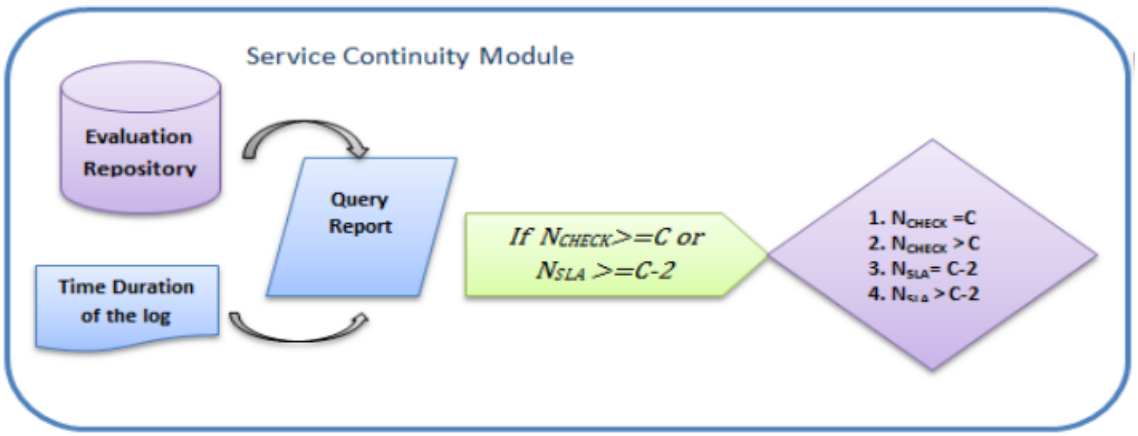


Figure 8.1: *Service Continuity Framework* by S. Khan, 2022.

Figure 8.1 provides an overview of the proposed service continuation framework. The proposed framework approach starts with the evaluation repository (the outcome of our previously developed framework is shown in Figure 7.1). We also introduce the time window concepts and assume that the time window is predefined during the SLA negotiation and construction stage. In this framework, a time window or duration illustrates a predefined duration of time when the intermediate agent will operate the proposed service continuity framework during the interaction. We explain the time window concept in Chapter 6, which discusses our previous service management framework. We consider the framework starting to operate from the midpoint of the interaction and close to the endpoint of the interaction as the best practice. Moreover, the evaluation repository is constructed using a predefined time window that is explained in Chapter 7.

The evaluation repository is used to develop the queries in the query report module in the next step. This module uses various rules and query approaches to retrieve the outcome. The module also summarizes how to reach the threshold outcome from the evaluation repository. In the next step, we determine the maximum value of acceptance and develop the rules. Finally, we employ the maximum value of the acceptance rule and obtain the outcomes, which is the decision that the model makes. As shown in Figure 8.1, the proposed framework consists of three stages as follows:

Stage 1: Summarisation of reaching the threshold

In the previous framework (condition evaluation and violation prediction framework), the intermediate or neutral evaluation agent interacts with both parties (service consumer, service provider) and collects all the related information regarding the deliverable services, and then engages the checkpoint thresholds (critical checkpoint and SLA checkpoint threshold). The agent applies the rules to evaluate how the QoS operates regarding the utilised threshold. The summarisation of reaching the threshold stage of the service continuation framework considers the results of the previous framework and is analysed further. The details of the approach are described in the next section.

Stage 2: Employing the maximum count of acceptance rule

In this stage, the intermediate or neutral evaluation agent performs another stage of the evaluation process in terms of the condition for a maximum count of acceptance rules predefined and committed to in the SLA. This stage aims to analyse, identify, and employ the acceptance value against all the service discrepancies that may or may not lead to a service-level contract violation. The details of employing the maximum count of acceptance rules are described in the next section.

Stage 3: Evaluation and decision obtained.

In this stage, the intermediate agent evaluates the results obtained from stage 1 against the stage 2 rules and obtains a decision on whether the service consumer should continue their business transaction with their current provider or should consider engaging a new provider to support and deliver the services according to their business needs. The details of stage 3 are described in the next section.

The service continuity framework comprises these three stages. We have provided an overview of these stages and discuss each comprehensively in the following sub-section.

8.2.1 Stage 1: Summarisation of Reaching the Threshold

Chapter 7 details a service evaluation and violation prediction framework, which develops an evaluation repository by performing a range of processes. The evaluation repository consists of the data or a record of the service performance evaluation data against the critical checkpoint threshold and SLA checkpoint threshold. After evaluating the service performance, we have labelled the traffic as follows:

1. If the performance of the current service is > the critical checkpoint threshold, there is no need to be labelled.

2. The performance of the current service is < the critical checkpoint threshold, then labelled as N_Check
3. The performance of the current service is = the critical checkpoint threshold, then labelled as N_Check
4. The performance of the current service is > the critical checkpoint threshold but < SLA checkpoint threshold, then labelled as N_Check
5. The performance of the current service is > the SLA checkpoint threshold, then labelled as N_SLA

Our proposed service continuity framework access evaluation repository consists of a list of the labelled services, which varies according to the aforementioned conditions. We need to summarise the count of the labelled (N_Check) services to determine how many times services reach the critical checkpoint threshold. Similarly, we determine how often the services reach the SLA checkpoint threshold by summarising the count of the (N_SLA) labelled services.

At the end of this stage, we have a summary or count of the N_Check and N_SLA services. This information is used for the next stage

of the proposed framework and the maximum count of the accepted framework. The approach details are given in the following section.

8.2.2 Stage 2: Employing the Maximum Count of Acceptance Rule

In this stage, firstly, the intermediate agent ascertains a value for acceptance. The acceptance value can be defined as the acceptance of the maximum number of times the performance is labelled as N_Check in a certain duration. Thus, we assume that the SLA's acceptance value is predetermined and agreed upon by all interacting parties and documented in the SLA contract. For this research, we define the maximum acceptance count as 5 for N_check labelled services and 3 for N_SLA labelled services.

We obtain the summary or count of N_check and N_SLA services from stage 1. We already know that the N_check and N_sla labelled services information is obtained from our runtime network monitoring traffic and assumes that the service provider delivers the services. After ascertaining the value of acceptance and the summary or count of N_check and N_SLA, the model analyse them and perform stage 3 to evaluate the rules and obtain the decision.

8.2.3 Stage 3: Evaluation and Decision Obtained.

The primary decision-making process is conducted in this stage. To implement the decision-making process, we develop some 'if-then' conditions and employ the condition in the framework. The conditions that we developed are as follows:

As previously mentioned, the count of acceptance is specified as C = 5; therefore, the rules are:

1. If the count of N_check = C, then condition 1 will be initiated.
2. If the count of N_check > C, then condition 2 will be initiated.
3. If the count of N_sla = C -2, then condition 3 will be initiated.
4. If the count of N_sla > C-2, then condition 4 will be initiated.

The implications of the conditions are as follows:

1. Condition 1 depicts the decision: The service is at risk, and proactive action is required.
2. Conditions 2 and 3 depict the decision: The service is tending toward an SLA violation.
3. Condition 4 depicts the decision: The SLA has been violated.

The framework or model employs these four if-then conditions and their implications. As seen from the above framework discussion, the frameworks are interdependent, which means that the outcome of every stage is required to operate for the following stage. The model assesses the current performance of the services by considering that each stage of the model depends on the completion of the previous stage. Finally, the model produces a predictive decision that assists all interacting parties, primarily the service provider, in understanding the overall performance of the deliverable services and acting accordingly. In addition, the service consumer also has a very transparent understanding and can make the formed decisions

regarding the continuation of the current and influence future contracts.

8.3 Experiments and Results

The previous section detailed the service continuation decision-making framework comprising three stages. Each stage has a progressive approach to implementing and integrating them with other approaches. These stages also need to be integrated with our previously developed frameworks to validate the success of the proposed SLA-based framework in terms of achieving the QoS. To implement the proposed framework, we engineered the system using the following tools.

1. Python programming language: A general-purpose, high-level, object-oriented programming language to facilitate the programmer's development of clear, logical code for projects of various scales. The primary philosophy of choosing the python programming language for its code reliability in dynamically typed and multiple programming paradigms.

2. Jupiter Notebook is used to write and run the code and receive the results.

This is a more straightforward framework than the other four frameworks proposed in this thesis. The framework's implementation uses the Python language. The framework primarily uses the service performance information from the previous framework and then applies conditions and obtains the outcome, which we refer to as informed predictive decision-making by the developed model.

In this section, we discuss the implementation process of the framework and demonstrate the results, followed by discussions. The implementation process comprises three stages, which are detailed in the following sub-section.

8.3.1 Implementation

To achieve the objective of SLA approach-based QoS delivery in service-oriented SDN, we proposed an SLA lifecycle in this thesis. The SLA lifecycle that we developed comprises various frameworks; as a result, we have developed and implemented the SLA negotiation framework (chapter 5), service management framework (Chapter 6), service condition evaluation and violation prediction framework (Chapter 7), all of which are interconnected. The last module of the SLA lifecycle is the service continuation framework. As the SLA lifecycle modules are interconnected, implementing one framework depends on the outcome of another.

The proposed service continuity decision-making framework implementation process initiates from the outcome of the previous framework (service evaluation and violation prediction framework discussed in Chapter 7). We have the QoS traffic dataset from the previous framework compared against the critical checkpoint threshold and SLA checkpoint threshold and labelled accordingly. Hence, the labelled QoS traffic dataset is considered the primary element for the proposed service continuation framework implementation.

Firstly, to implement the framework, we summarise the count of all the labelled datasets to ascertain how often the service performance has reached the pre-configured critical checkpoint threshold. Similarly, we ascertained that the count of the labelled services reached the SLA checkpoint threshold. We used the Python programming language to engender the code and produce the outcome of the count of all labelled services. The outcomes are demonstrated in the results section.

In the second stage, we employed the maximum count of the acceptance condition. Let us assume that the maximum acceptance count is 5, predetermined during the SLA negotiation stage and agreed upon by all interacting parties. Finally, we deploy the evaluation condition into the framework or model and predict the decision. The evaluation conditions described in the framework section are implemented using the Python program code. The outcomes are demonstrated in the result section, followed by a discussion of the framework.

8.3.2 Results

This section details the results of implementing our service continuation decision-making framework. The implementation process is designed to achieve objective 4, validating and ascertaining the efficacy of an intelligent decision-making framework for service continuity. The evaluation compares the current network service performance variations and QoS level with the SLA expectation. Moreover, the evaluation analyses the efficacy of our SLA) management framework in SDN for informed decision-making. In this section, firstly, we discuss the results of the service continuation decision-making framework, followed by discussing the achievements of the framework. The results are described as follows:

Step 1: Results of summarisation of reaching the threshold.

Using the dataset, we performed this task to determine how the developed model performed with different QoS traffic data. Figure 8.2 shows the labelled or significant traffic data input results that reached the applied threshold and illustrated the applied condition as 'True'. Figure 8.3 shows the count of the labelled traffic that makes the applied condition 'True'.

ts	bandwidth	throughput	jitter	packet_loss	delay	n_sla	n_check
0	25638.333333	17136.833333	69.923333	3.160000	375.400000	False	True
1	27089.116667	17633.383333	62.513333	2.771667	319.133333	False	True
2	25026.766667	16066.566667	53.200000	2.998333	267.583333	False	True
3	24119.283333	15195.600000	58.033333	3.053333	338.666667	False	True
4	27373.466667	17731.766667	67.513333	2.835000	353.466667	False	True
5	24679.716667	15147.250000	63.151667	2.890000	371.183333	False	True
6	27110.633333	16469.016667	66.193333	2.800000	342.333333	False	True
7	25142.283333	15751.983333	58.951667	3.198333	315.600000	False	True
8	26781.550000	16573.400000	63.828333	3.178333	324.750000	False	True
9	25207.650000	16846.133333	65.590000	3.051667	343.950000	False	True
10	23900.233333	15297.900000	51.598333	3.391667	273.616667	False	True
11	21321.133333	13534.266667	73.776667	2.953333	443.450000	False	True
12	24912.100000	14888.750000	70.515000	3.415000	440.616667	False	True
13	23556.366667	16226.450000	67.408333	2.845000	406.683333	False	True

Figure 8.2: labelled traffic dataset

```
n_check: 24
n_sla: 0
```

Figure 8.3: Count of labelled traffic dataset

Step 2: Results of evaluation and decision obtained.

Figure 8.4 shows the predictive decision that our developed framework makes by analysing and evaluating the current performance of the services in all the previously described evaluation conditions.

```
Result condition: 2
Service has a tendency of violation
```

Figure 8.4: The model's predictive decision

8.3.3 Discussions

The experiment results show that the proposed service continuation decision-making framework is able to evaluate the current performance of the SOA-based SDN. Moreover, the framework assists in predicting the possibility of a forthcoming violation of the SLA to help the service provider take appropriate action to preserve the SLA before the violation occurs. Furthermore, in the discussion, we argue that this makes the framework more functional, reliable, and efficient enough to serve the purpose.

1. The experiments to evaluate the proposed framework considers two checkpoint thresholds to ascertain the service performance. As previously discussed, the framework enables the application of multiple checkpoints based on business demand.

2. The value of maximum acceptance of the value of C is flexible and enables acceptance of the values as long as the value can relate to the framework and SLA contract.

3. Similar to the service evaluation and violation prediction framework, the discussion of the proposed framework and the experiment show that the framework can successfully evaluate the service performance from a runtime system and generate a predictive decision regarding service continuity decision-making. Moreover, the framework can detect SLA violations and predict the possibility of a violation of the SLA contract, assisting in preserving the SLA. Thus, the framework prevents service providers and consumers from incurring penalties and facing additional aggravation.

4. The system we have developed can perform with a dynamic range of values according to business and SLA demand. This option opens up the opportunity to use the developed system in SDN and various other networks, such as SOA-based military tactical networks and heterogeneous radio networks.

The above discussion shows that introducing advanced features that enable the customisation of the proposed service continuity decision-making framework enables its successful deployment in various SOA-based SDN-oriented architectures. Therefore, we successfully achieved objective 4 to develop an intelligent decision-making framework for service continuity.

8.4 Conclusion

This chapter presents an impartial and trustful service continuity decision-making framework in SOA-based SDN to facilitate the impartial association of service continuity decisions if service reliability or confidence is achieved. A third party carries out the framework in the presence of the consumer party during the interaction, especially at the midpoint of the

interaction upon which all the interacting parties agreed. The fundamental hypothesis of our SLA-based framework is that a trust relationship has already been established during the interaction, and the trust relationship is sustained by consistently monitoring, examining, and evaluating the interaction. Thus, this framework assists in decision-making by performing an impartial assessment of the outcome of the *service evaluation framework* (Chapter 7), where the evaluation framework enables the performance to be measured using the service performance evaluation and ensuring that the services are operating with completeness, honestly, and securely.

The framework comprises three stages: stage 1: summarisation of reaching the threshold, stage 2: employing the maximum count of acceptance rule and stage 3: evaluation and decision obtained. The implementation and validation results show that our proposed framework successfully made intelligent, data-driven service continuation decisions in a neutral position from the impartial evaluation of the service performance of the deliverable services. This framework assists in preserving service reliability and transparency and the scope for the re-adjustment or re-calibration of the trust relationship. Therefore, we can conclude that this framework can offer a genuine representation of the outcome of the objective of delivering reliable services and preserving the trust relationship with possible re-adjustment.

8.5 References

Staab, S., Bhargava, B., Leszek, L., Rosenthal, A., Winslett, M., Sloman, M., Dillon, T. S., Chang, E., Hussain, F., & Nejdl, W. (2004). The pudding of trust. *IEEE Intelligent Systems, 19*(5), 74-88.

Chapter 9

RECAPITULATION AND FUTURE WORK

9.1 Introduction

Service-oriented architecture (SOA) has attracted widespread attention from the IT industry and researchers. A trust relationship is a key element in successful service-based business transactions. The lifecycle of trust modelling comprises trust building, maintenance and the trust decline stage, which is discussed in detail in Chapter 4. Reputation data-driven trust modelling is an area that is generating great interest in the field of service-oriented business. Ensuring QoS and trust are essential in a service-oriented business environment, while reputation rating plays a significant role in building a trust relationship with an unknown entity. Ensuring QoS in a service-oriented climate is not a new research issue; however, QoS in a service-oriented SDN is a very immature research area, evident from our state-of-the-art literature survey detailed in Chapter 2.

Several solutions are proposed to achieve QoS in SOA-based architecture and SDN architecture to address this issue. However, none of these solutions proposes an integrated or combined solution that preserves trust relationships by ensuring uninterrupted QoS delivery through continuous service performance monitoring and impartial evaluation of the deliverable services and assists in service performance-based decision making building a trust relationship. Moreover, no integrated approaches can predict possible service contract violations and save the organisation from significant financial loss.

To mitigate the research issues mentioned above, this thesis proposes complete solutions that evolve in six research directions and offers an intelligent SLA-based QoS guaranteed framework for informed decision-making in service-based SDN.

This chapter discusses the pressing issues related to ensuring QoS delivery in service-oriented SDN architecture addressed in this thesis which is presented in section 9.2. Section 9.3 outlines and discusses the significant contributions that this thesis makes to the existing literature. Finally, the thesis concludes by outlining future research directions in section 9.4

9.2 Problems addressed in this thesis

This thesis addressed the following six significant issues associated with ensuring QoS delivery and maintaining trust relationships in a service-oriented SDN environment. It defines the concepts of trust, trust maintenance, trust-based relationships or interaction, the trust evaluation lifecycle model, reputation data/rating, service reputation, quality of service (QoS), trust in a business context, and trust in a service-oriented environment.

- It proposes a framework for personalising and negotiating the service requirements. This framework enables two interacting parties (the service provider and consumer) to reach an agreement called a service-level agreement (SLA) in the presence of a third party or intermediate agent after negotiating the essential service requirements. The framework involves various activities, such as articulating the service requirements, negotiating the service requirements that include complex or conflicting service requirements, selecting a suitable service provider to deliver a particular service request and assisting in decision-making in relation to accepting the service request. The framework also provides direction on constructing an SLA from the outcomes of the negotiation process.

- It proposes a framework for service management that includes real-time proactive, continuous performance monitoring of deliverable services. The service management framework comprises passive performance monitoring, including network traffic type identification, classification, and prioritisation. Therefore, the service management framework is a conjoint approach used to proactively monitor the performance, analyse the performance, and produce an overall performance outcome during the time-space of interaction.

- It proposes a framework for service performance evaluation and predicts a possible SLA violation tendency. This framework is used for measuring the quality of the services being delivered against the range of services agreed upon in the SLA. This

framework is also used for preserving the SLA by predicting a possible violation to assist the service provider in taking early measures to avoid SLA violation.

- It proposes a framework for service continuation decision-making based on the previous behaviours of the service provider during the time-space of interaction.

- It validates the proposed methodology for proof of concept by simulating the framework and experiments.

9.3 Contributions of this thesis to the existing literature

The significant contribution of the thesis in terms of the existing literature is that this thesis proposes a complete methodology for ensuring personalised QoS delivery and assists in maintaining trust in SOA-based SDN architecture. The complete solutions of the SLA lifecycle-based QoS delivery encompass four distinct stages, and each stage consists of a couple of frameworks that significantly contribute to the existing literature. The frameworks are as follows:

1. A service negotiation framework for personalised service delivery in SDN. This framework comprises two significant approaches, one is reputation data-driven service provider selection, and the other one is reputation data-driven decision making which assists the service provider in evaluating the suitability of accepting a service request.

2. A service management framework for managing deliverable services using a conjoint approach comprising proactive, continuous, and passive performance monitoring. This is a framework to proactively monitor the performance of the deliverable services during the time-space of interaction to maintain the SLA contract and the trusting relationship between the two parties.

3. A framework for continuously evaluating service delivery performance and predicting the possibility of service contract violation instead of a one-off service performance evaluation. This framework assists in the impartial review of service performance conducted by a third-party agent to preserve the SLA by involving an early intervention if the service performance is degrading.

4. A framework to assist the service consumer in terms of service continuity and intelligent decision-making based on the performance of the delivered services during the time-space of interaction. This framework enables the service provider to understand the area that requires adjustment to preserve the SLA and maintain the trust relationship.

Furthermore, this thesis proposes a conceptual realisation of the relationship between QoS, service reliability and trust evaluation. In addition, this thesis discusses the importance of ensuring QoS to ensure reliability which is a primary component of the trust relationship for a service-oriented business such as an SOA-based SDN environment.

This thesis provides a complete lifecycle-based QoS solution for service-oriented SDN, which is an additional contribution to the existing literature on QoS in the SDN and service-oriented environment research domain. Finally, comprehensive experiments were conducted, the proposed SLA lifecycle-based framework is validated through simulation-based experiments and the outcomes are reported in the thesis. In this section, we give a brief overview of all the contributions made by this thesis to the existing literature.

Contribution 1: Current State-of-the-art Literature Survey

Chapter 2 gives an extensive review of the existing approaches to QoS in SDN architecture in a service-oriented environment. It broadly categorised the relevant studies into the following five groups:

1. QoS-based controller design,

2. Resource allocation-based approach,
3. Queue scheduling and management-based approach,
4. QoS-driven optimal routing, and
5. SLA is based on quality management in SDN.

In addition, we compare the working of these techniques against the identified requirements of guaranteeing end-to-end QoS provisioning in SOA-based SDN architecture and present directions for future research. To determine the critical requirements in ensuring end-to-end QoS provisioning in SOA-based SDN, we take inspiration from cloud computing, one of the successful computing architectures that have arisen from SOA. This thesis focuses on how cloud services between users and providers are formed. In this process, researchers have identified many criteria that range from certifications to migration support, which must be considered when deciding on a service.

We argue in this thesis that if an SOA-based SDN needs to provide a network that satisfies the requirements of different services, it must follow a similar approach that first ensures the correct service and then manages the service to ensure that the expectations are achieved. In addition, we determine that the following requirements need to be met to achieve such an approach that can guarantee end-to-end QoS provisioning in SOA-based SDN architecture.

1. Ability to personalise the QoS required from a service
2. Reputation value-based network service selection
3. Measuring the satisfaction of a network service provider based on the QoS it provides
4. Measuring the network provider's actual QoS delivery by considering its composite nature.

We reviewed all the mentioned categories in the existing literature, comprehensively described their position in terms of the aforementioned requirements and undertook a comparative analysis. Furthermore, the objective of the literature review, which is to identify the problems in the current literature concerning QoS in SOA-based SDN architecture, is identified in this chapter. The comprehensive survey-based literature review also identifies the issues we address in this thesis. This chapter illustrates that the problems we address in this thesis have not been previously addressed and resolved in the literature.

Contribution 2: Definitions of trust, reputation, QoS and their related concepts for a service-oriented environment

The existing literature does not clarify the concepts related to the service-oriented environment. Moreover, the existing literature fails to describe the relationship or dependency between the primary elements such as reputation rating, QoS and reliability for a trust-based relationship in a service-based business environment. This thesis identifies these gaps and clarifies the importance of QoS in a service-oriented environment that helps to achieve reliability between the service consumer and service provider, on which the trust relationship is founded.

Another significant contribution of this thesis is further exploring and orienting the trust evaluation model in our approach in Chapter 4, where the three stages of the model consist of trust building, trust maintenance and trust decline. This thesis successfully expands the trust evaluation model, starting from trust building to the trust decline stage. To the best of our knowledge, this thesis is the first attempt to provide a complete framework that can operate in every stage of the trust evaluation model.

Contribution 3: The methodology for personalised service negotiation for QoS delivery is to build a trust relationship in an SOA-based SDN.

The third significant contribution of this thesis is the proposal of a service negotiation framework to formulate and negotiate the service requirements to construct an SLA. This framework is used to determine the service requirements in the presence of a third party or intermediate agent and comprises a five-stage process resulting in the service requirement request becoming an SLA contract between the service provider and the service consumer. The constructed SLA can then be used as a standard process and benchmarking standard for service monitoring and evaluation that impacts trust management and helps the decision maker decide whether to continue the SLA with the current QoS settlement or re-adjust at the end of the

interaction. Chapter 6 comprehensively discusses the service negotiation framework. To the best of our knowledge, the existing literature does not propose any framework that enables personalised service requirement formalisation and negotiation in an SOA-based SDN, which contributes to the trust-building stage of the trust evaluation model. The key features of this framework are as follows:

1. The framework presents a process flow that assists in formalising the service request and proposes a template by which both parties can articulate their service requirements in a structured manner with the assistance of the third-party or intermediate agent. This formal template and the quantifying service requirement process help the service negotiation process to reach an SLA.

2. The framework presents a process flow and a template for formalising the proposal from the service providers. The service description template (SDT) assists in translating the service request to an SDT. This template assists in identifying and addressing any conflicting or unfeasible requirements and solicits proposals from the prospective service providers' ambiguous responses if present.

3. A method is proposed that assists the service provider evaluate the suitability of accepting a service request. This method considers three factors to determine the suitability of accepting a service request.

4. A reliable and data-driven reputation rating approach is proposed in this framework, where the reliability of the consumer company is considered the reputation rate of the consumer company and the reputation rate feedback of the other providers who have worked with this consumer company before.

5. A fuzzy association metrics-based suitability calculation is proposed in this framework, where the calculation is performed by considering the factor strength. In our approach, of the three factors, we consider reliability as having the highest strength and the service duration factor as having the lowest strength.

6. A method is proposed that assists the service requester in selecting the most suitable service provider. This proposed method has two approaches, one for selecting a suitable

service provider from a range of existing service providers and the other for emerging service providers with no reputation rating. One of the significant contributions of this method is proposing a reputation rating data-driven service provider selection in SOA-based SDN.

7. A synthetic time series reputation rating dataset is developed in this framework and validated using various intelligent approaches. This reputation rating dataset is a significant element and can be reused for future research.

Contribution 4: Methodology to enhance service management to ensure QoS delivery and maintain a trusting relationship in SOA-based SDN

The fourth significant contribution is that this thesis proposes a framework for continuous monitoring of the deliverable services to manage the services. This framework is comprehensively discussed in Chapter 7. To the best of our knowledge, the existing literature does not propose any framework for managing services using a conjoint monitoring approach that considers the dynamic nature of trust, the dynamic nature of both interacting parties' performance and the characteristic features of the trust maintenance stage. The features of this framework are as follows:

1. It proposes a method by which both the interacting parties (service provider and consumer) and third-party agents can obtain real-time performance information.

2. It proposes a method for real-time proactive, continuous performance monitoring instead of one-off performance monitoring that assists in preserving the SLA contract that is established in the service negotiation stage.

3. It presents a mechanism to ascertain the number of time windows, time slot(s) and number of checkpoints during the interactions. The number of time slot(s) is designed to evaluate the performance of the deliverable services continuously and proactively.

4. It proposes a method for passive monitoring to manage the services. Passive monitoring assists the monitoring agent in analysing the current service performance, and based on the monitoring outcome, it identifies the vulnerable areas and pays more attention to them.

5. This framework proposes a machine learning-based approach for network traffic identification or traffic recognition.

6. The framework is validated using various intelligent tools such as simulation, machine learning, and Python.

Contribution 5: Methodology to undertake impartial service evaluation and violation prediction for trust relationship preservation in an SOA-based SDN environment

The fifth significant contribution of this thesis is that it proposes a framework for service evaluation and violation prediction to deliver impartial and trustful services. The framework details are discussed in Chapter 7. To the best of our knowledge, the existing literature does not propose a service performance evaluation framework that can operate impartially and predict a possible violation before a violation takes place. We validate the framework using two stages; a) the formulation of the quantifiable services and b) the condition evaluation services. We validate the first stage using various steps, including collecting a time series dataset using the Zabbix monitoring tool and extracting the data in .csv format, cleansing the dataset and preparing the dataset according to the evaluation metrics. In addition, we validate the second stage using several steps, including employing the service degradation critical checkpoint threshold and SLA threshold using the Zabbix monitoring tool. In addition, the Python programming language is used to evaluate service performance with the critical checkpoint threshold and SLA threshold, followed by implementing the performance discrepancy condition and labelling the traffic using the Python programming language. The salient features of this framework are as follows:

1. The proposed framework can successfully evaluate the service performance from a runtime system.

2. The proposed framework can detect SLA violations and predict the possibility of SLA violations which assists in preserving the SLA.

3. The proposed framework prevents service providers and consumers from incurring unpredicted penalties, which may result in significant financial loss and additional aggravation.

4. The proposed service evaluation and violation prediction framework experimentation considers four commonly used performance parameters to evaluate the network service performance. However, the option is open to introducing various other performance parameters or additional parameters in the framework to assess service performance.

5. The proposed framework employs two checkpoints. However, according to the business strategies, the framework considers generic options for employing additional checkpoints.

6. The proposed framework is open to applying the critical checkpoint threshold value and SLA threshold values according to consumer expectations and the SLA contract.

7. The proposed framework has the ability to perform with a dynamic range of values according to business and SLA demand. This option allows the use of the proposed system in SDN and various other networks, such as SOA-based military tactical networks and heterogeneous radio networks.

8. We validate the framework's effectiveness by comparing the simulated network use performance for our proactive, continuous performance monitoring framework.

Contribution 6: Methodology to facilitate impartial service continuity intelligent decision-making in SOA-based SDN environments.

The sixth significant contribution of this thesis is that it proposes a framework for impartial service continuity and intelligent decision-making in SOA-based SDN environments. It is important to note that the service continuity framework is carried out at the end of the interaction. To the best of our knowledge, the existing studies do not propose any framework that assists in service continuity decision-making and re-adjusting the trust relationship in interactions that occur during the trust maintenance stage. The framework details are discussed in Chapter 8. The significant features of this framework are as follows:

1. The proposed framework considers agents' static and dynamic behaviour during the interaction.

2. The proposed framework experimentation considers two checkpoint thresholds to ascertain service performance. The framework enables the application of multiple checkpoints based on business demand.

3. In the proposed framework, the value of the maximum acceptance of the value of C is flexible and enables acceptance of the values as long as the value can relate to the framework and SLA contract.

4. The proposed framework can successfully evaluate service performance from a runtime system and generate a predictive decision regarding service continuity decision-making.

5. The proposed framework can detect SLA violations and predict the possibility of SLA violations, assisting in preserving the SLA

6. The proposed framework prevents service providers and consumers from incurring penalties and additional aggravation

7. The proposed framework can perform with a dynamic range of values according to business and SLA demand. This option allows the use of the proposed system in SDN and other networks, such as SOA-based military tactical networks and heterogeneous radio networks.

Contribution 7: Reputation rating synthetic time series dataset for service consumers and service providers.

Using the SLA-based framework proposed in this thesis, we also developed a reputation rating time series dataset for service consumers and providers. Due to confidentiality reasons and the current internet-based or service-based business environment, it is almost impossible to find a real-life reputation rating dataset. Therefore, researchers face difficulty conducting experiments and validating their research due to the lack of a validated dataset. To the best of our knowledge, this is the first synthetic time series dataset developed regarding the reputation rating of an organisation. The development and validation details of the reputation rating synthetic time series dataset are comprehensively discussed in Chapter 5, as this dataset is used in the experiments with our proposed service negotiation framework. The significant features of this reputation rating dataset are as follows:

1. The popular Euclidean distance approach is used to develop the transaction trend feature of the dataset.

2. The dataset consists of 500 records, meaning the dataset can be considered to be the reputation ratings of 500 service providers or service consumers.

3. We developed a time series dataset and performed a root mean square error (RMSE) analysis using the time series to determine the lowest error rate.

4. We developed the dataset using a fuzzy inference system (FIS) model.

5. The developed dataset consists of various essential features that can be used to validate future service-oriented or reputation-rating-related research.

Contribution 8: Strength-based fuzzy association rules for service request evaluation

We developed a strength-based fuzzy association rule and represented this in a multidimensional form, as reported in Chapter 5, section 5.4.2. This is a rule matrix of fuzzy interference systems (FIS) to map the input variables and receive output values that assist in intelligent decision-making. In this approach, a fuzzy association rule represents them in a multidimensional form. To the best of our knowledge, this is the first strength-based fuzzy association rule developed to evaluate the suitability of a service request. The development and validation details of the strength-based fuzzy association rules are comprehensively discussed in Chapter 5. The significant features of this approach are as follows:

1. The commonly used FIS is used to develop the strength-based fuzzy association rules with the help of the MATLAB tool.

2. We used the prevalent Mamdani method to calculate the provider's suitability and assist in critical, intelligent decision-making.

3. We developed the association rules by mapping the strengths of the multiple factors.

4. The proposed strength-based association rules can be used for other strength-based evaluation and decision-making purposes with various evaluation factors or metrics with multiple strengths.

9.4 Conclusion and Future Work

The work we have undertaken in this thesis has been published extensively as a part of the proceedings in peer-reviewed international conferences and international journals. We have

attached a list of selected publications, including one journal paper currently under review, one journal paper and four international conference papers published, and three journal papers that have been written and reviewed with the principal supervisor.

The standardisation and development of best practices for defining and managing SLAs in SOA environments is an active area of research due to the interest in dynamic discovery and the selection of services, the emergence of third-party services, and the desire for machine-readable SLAs with few or no humans in the loop (Bianco et al., 2008). Even though we have undertaken considerable research on the topic of this study, we realise that there is much scope for future work. We intend to continue working on this topic, primarily along but not limited to, the following lines.

1. The proposed SLA-based framework is a lifecycle model where each module has a dependency on its previous module. We develop a proof of concept (POC) to validate our analysis in this research. The POC is set on a small scale using various tools and techniques in a simulated environment. A future recommendation is to implement this research in a production environment with a broader context to maximise the benefits of this study.

2. Service-based SDN is an evolutionary concept that can leverage traditional networking into a service-driven network with prospective service-based prospects. The primary opportunities include personalised, differentiated service delivery, multi-agent service selection, service composition, multi-tenancy and more advanced features such as containerisation of the network segment with security. In this research, we viewed the SDN with SOA; however, due to limitations of timing and resources, we could not develop the concept's prototype. A future recommendation of this research is to introduce this concept to real-life production-based scenarios to maximise the benefits of SOA.

3. To develop the POC of the SLA lifecycle model in this research, we introduced various implementation tools and approaches to multiple environments where the outcome of one process is used for other methods. A future recommendation of this research is to develop the entire lifecycle model in a single environment where multiple research

approaches or technologies can operate in an integrated environment and create a single product of the SLA lifecycle model.

4. Process mining is an advanced technique to leverage the developed service formalisation and negotiation into the automated monitoring of the performance progress based on the agreed SLA.

5. A recognised standard for SLAs is needed for specification and management. SLAs need to be machine-readable to make the third generation of service-oriented systems possible—dynamic discovery, adaptation, and service composition. A significant contribution would be the creation of a generic and standardised framework that could provide the appropriate automation and support for (1) mapping contractual SLAs to standard and actionable implementations and (2) the monitoring and management of SLAs at runtime.

6. More research is needed to understand and determine the QoS of composite services. For example, if the performance measure for a set of services is known, how can the performance for a composite service that uses this set of services be determined?

9.5 References

Bianco, P., Lewis, G. A., & Merson, P. (2008). Service level agreements in service-oriented architecture environments.

Appendix A

(SERVICE MANAGEMENT)

1. Proactive Continuous Monitoring

Proactive continuous performance monitoring is the run-time network monitoring approach. In order to implement the approach, we have developed a simulation network using GNS3. The Zabbix monitoring tool is integrated with the simulation network to perform run-time monitoring of the network. The network simulation development consists of the following steps.

1.1 Windows 10 Virtual Machine

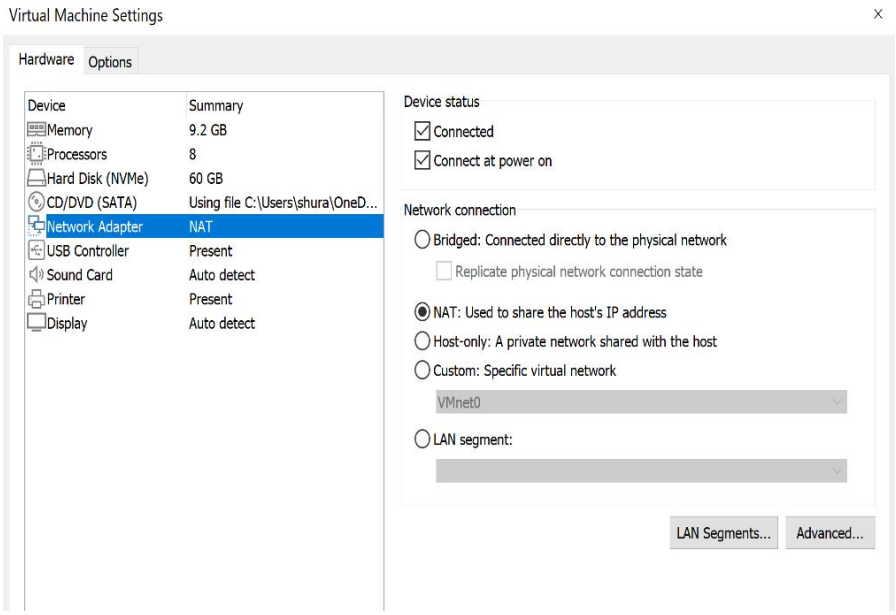


Figure A1: Windows 10 Virtual Machine configuration.

We have configured a windows 10 virtual machine to keep the whole simulation isolated from the rest of the applications of the computer. Figure A1 shows the virtual machine (VM) configuration used to build the machine.

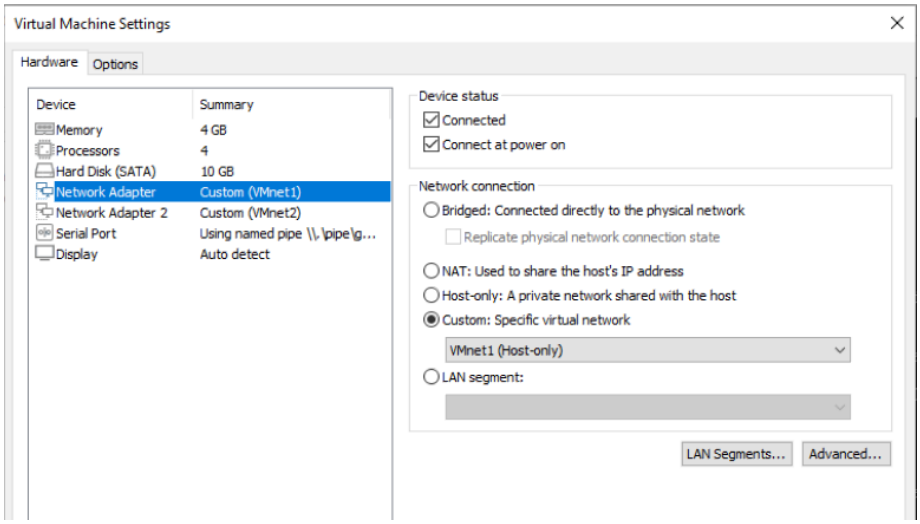


Figure A2: Windows 10 Virtual Machine network adapter configurations

We have activated two adapters in this VM named VMnet1 and VMnet2. One adapter is used to connect the virtual machine with the physical computer system, and another is used to connect the VM with the Zabbix monitoring agent VM that we will create inside the windows 10 VM. Therefore, the Zabbix monitoring agent VM operates as a netted VM. Figure A2 shows the VMnet1 and VMnet2 configurations.

1.2 Network Simulation Using GNS3

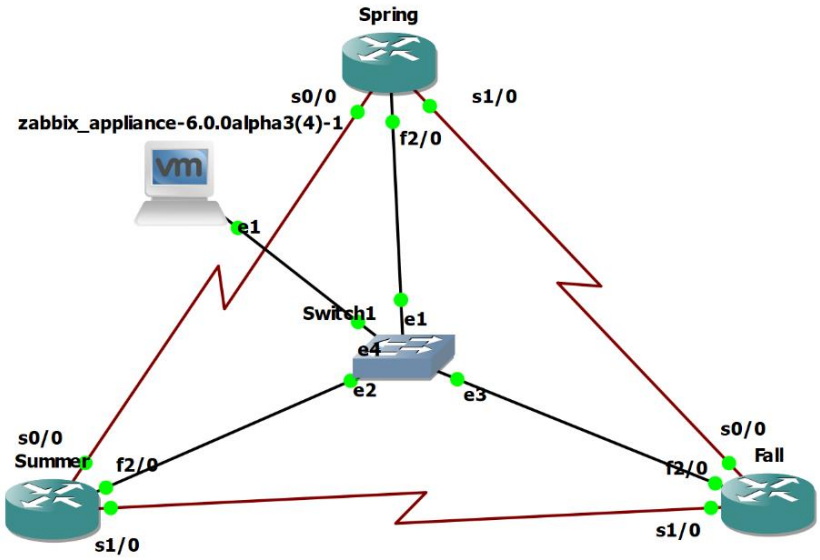


Figure A3: Network simulation structure using GNS3

We developed our run-time network using the GNS3 simulation and emulation tool. The structure of the prototype is shown in Figure A3 above. In order to develop the simulation, we used the c3640 router image. We have configured three c3640 routers in the above scenario, and the router is named Spring, Summer and Fall. We have configured a management switch to connect these routers. The successful Ping ensures the network is communicating with each other successfully demonstrated in figure A4

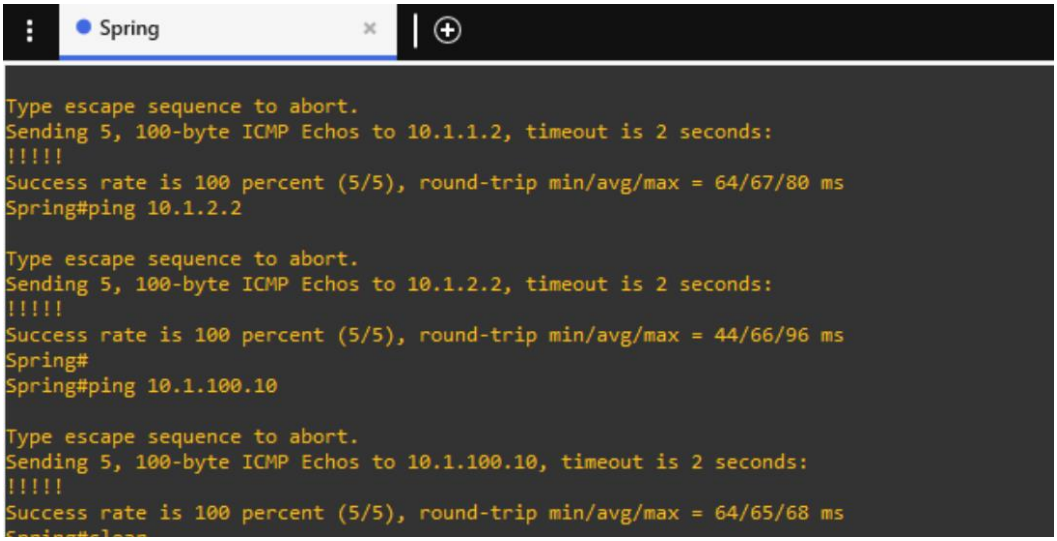


Figure A4: Successful ping reply from Spring router to all the other routers.

1.3 Zabbix Appliance for Network Monitoring

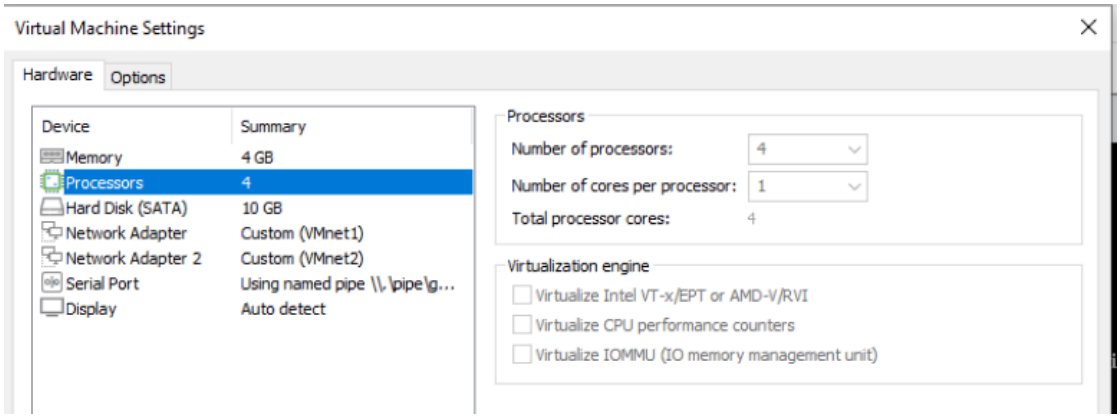


Figure A5: Virtual Machine (VM) for Zabbix appliance configuration.

Zabbix is the network monitoring tool used to monitor the network using a run-time monitoring approach. To configure the Zabbix monitoring, we have used the Zabbix appliance. We configure the Zabbix appliance in a virtual machine, and the configuration of the virtual machine is shown in Figure A5.

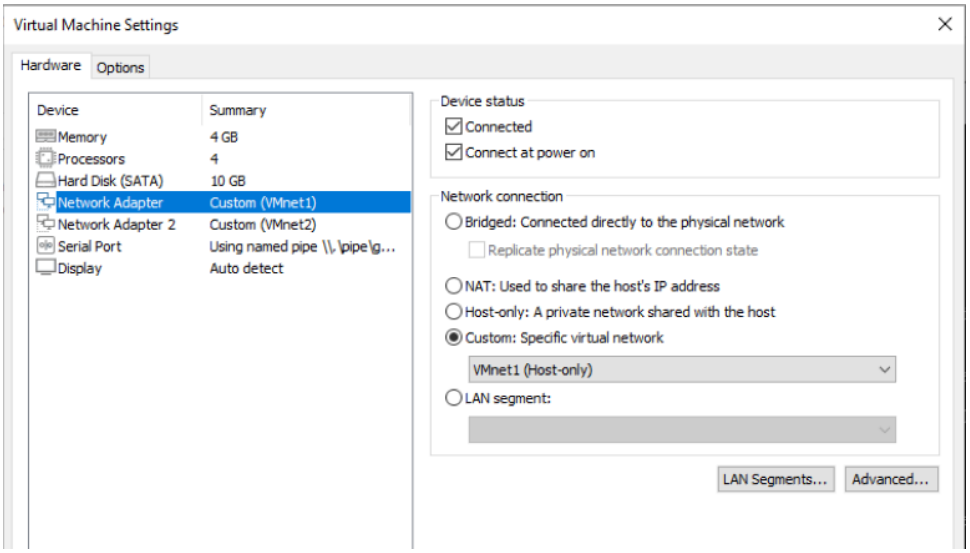


Figure A6: Virtual Machine (VM) for Zabbix appliance and VMnet1 adapter configuration.

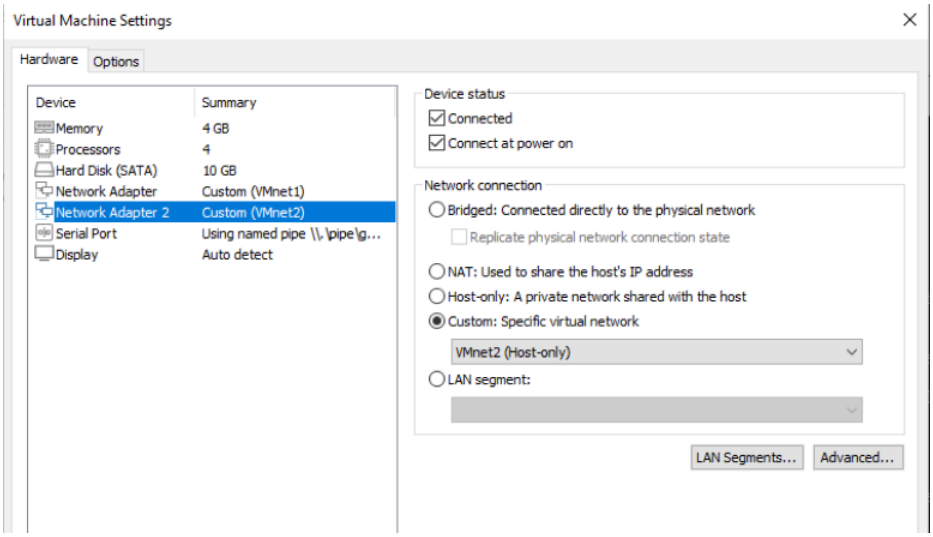


Figure A7: Virtual Machine (VM) for Zabbix appliance and VMnet2 adapter configuration.

In the Zabbix appliance VM, we have activated two network adapters. One adapter is used to access the Zabbix monitoring front end that can be accessed using a web browser, and the

configuration is shown in Figure A6. The other adapter is used to monitoring connect with the whole topology, and the configuration is shown in Figure A7.

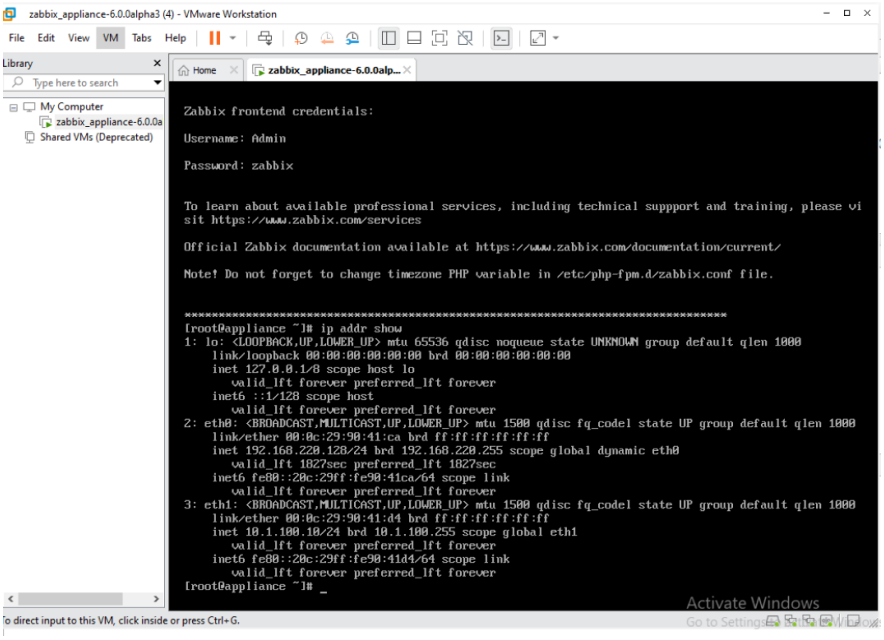


Figure A8: All of the active interfaces of the Zabbix monitoring agent.

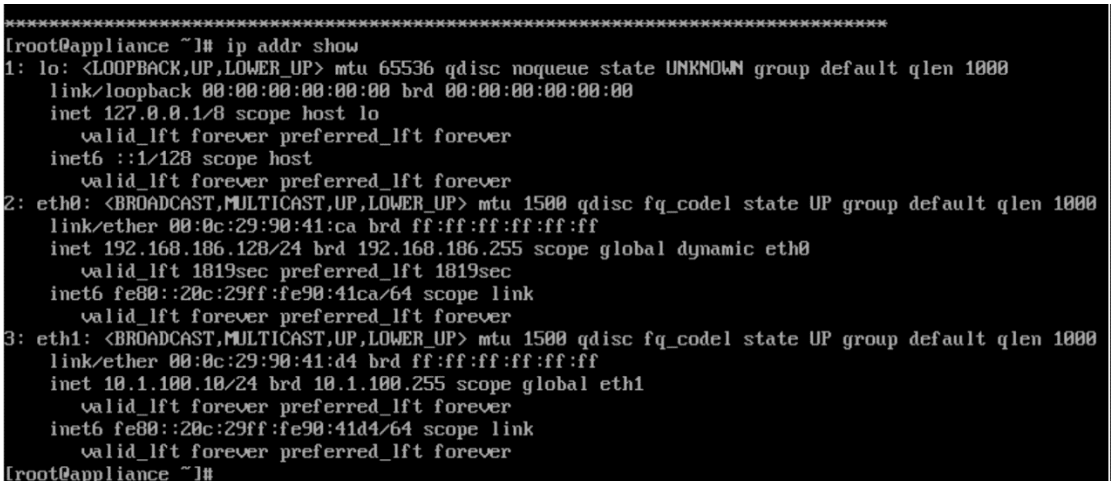


Figure A9: Closour look of all of the active interfaces of the Zabbix monitoring agent

Figure A8 and A9 demonstrate three interfaces named lo, eth0 and eth1. We configure the eth1 IP address as a static IP so that we can access the Zabbix monitoring dashboard using a web browser.

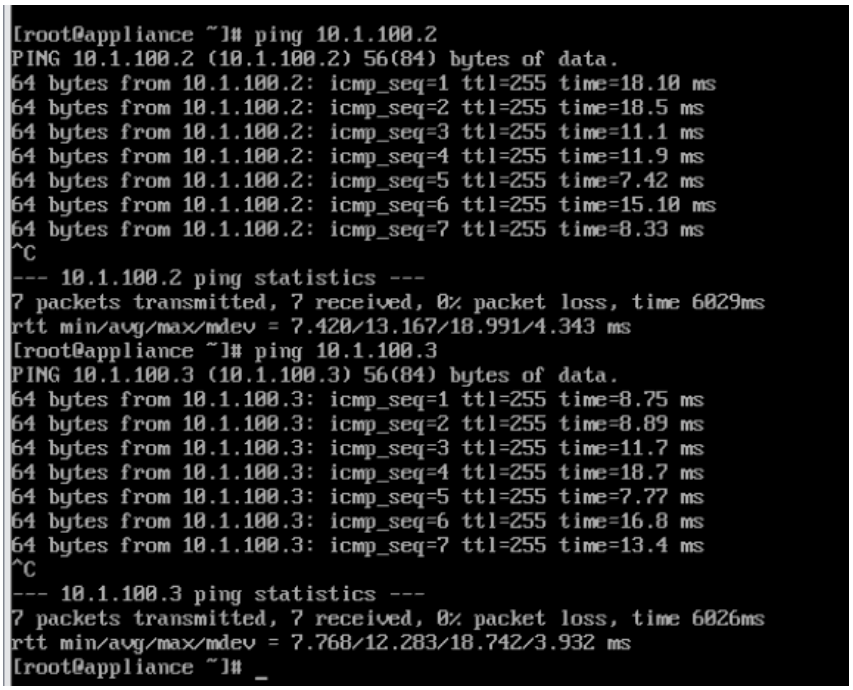


Figure A10: Confirming the connectivity with the routers of the topology using PING reply

We used the PING command to confirm the connectivity between the Zabbix monitoring appliance and the network routers. The successful ping reply demonstrates that the Zabbix monitoring agent is connected and communicating successfully. Figure A10 shows the successful ping reply from the router.

1.4 Zabbix Frontend Network Monitoring Dashboard:

As mentioned above, the Zabbix Frontend enables access to the network monitoring dashboard using the web browser. Figure A11 below demonstrate that the four devices are currently connected with the Zabbix monitoring tool.

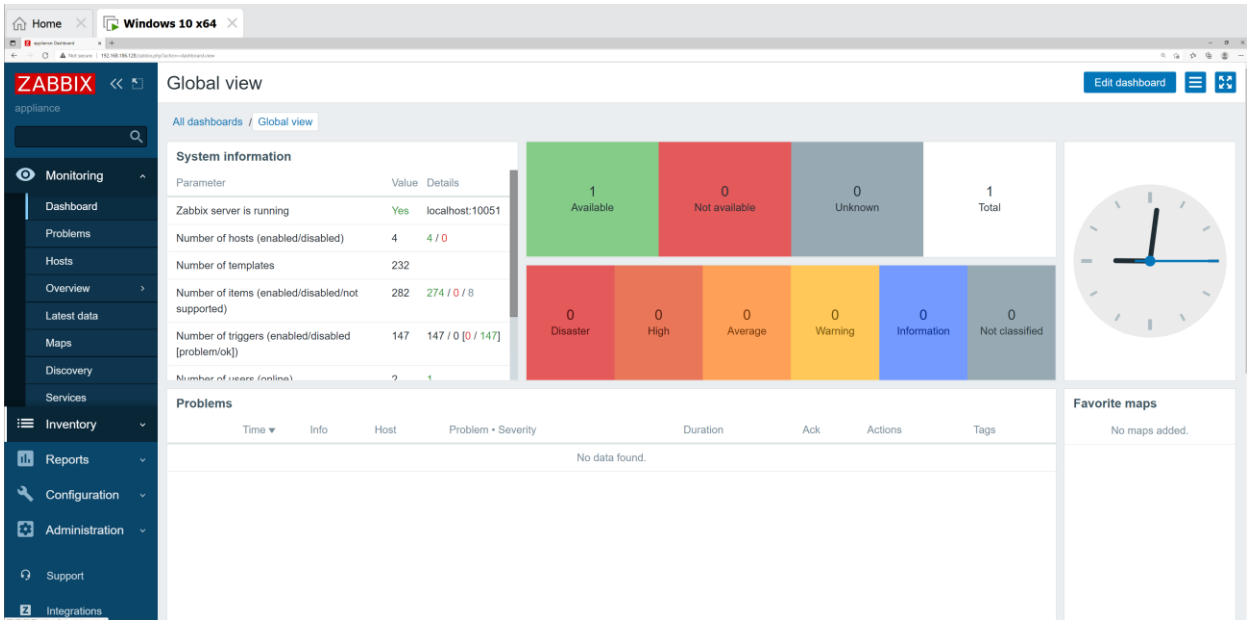


Figure A11: Zabbix monitoring dashboard

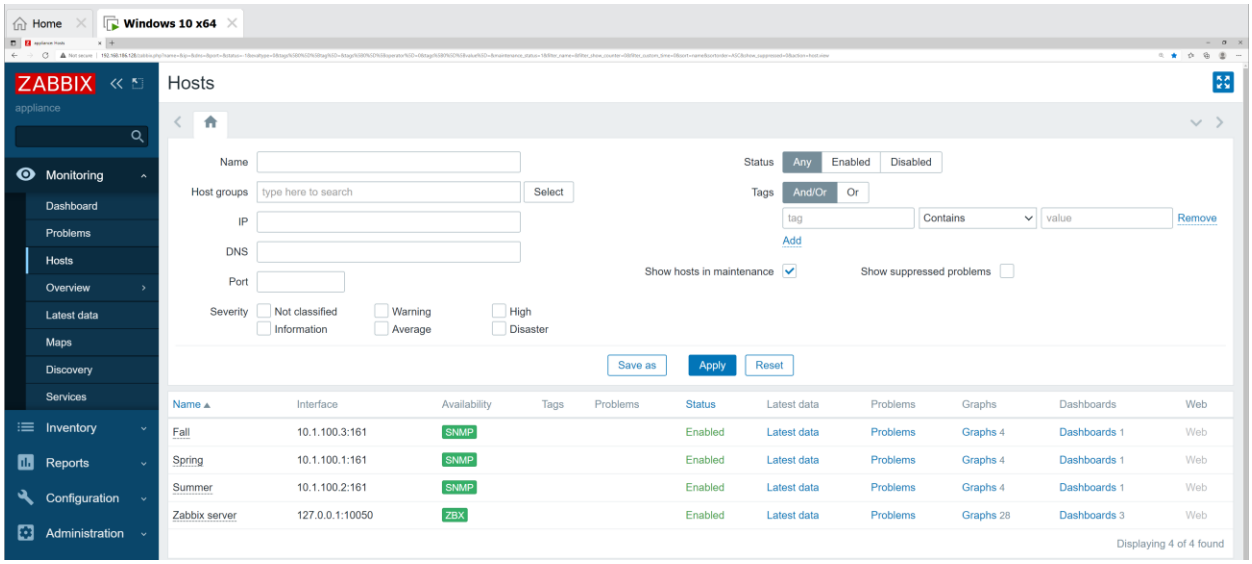


Figure A12: SNMP is activated on all of the connected devices

In order to activate the communication between Zabbix and the devices, we have enabled SNMP communication in all of the devices. Figure A12 shows that the devices and the Zabbix

server have the SNMP activated and enabled. Therefore, they communicate with each other, and Zabbix can access the devices and continuously monitor them.

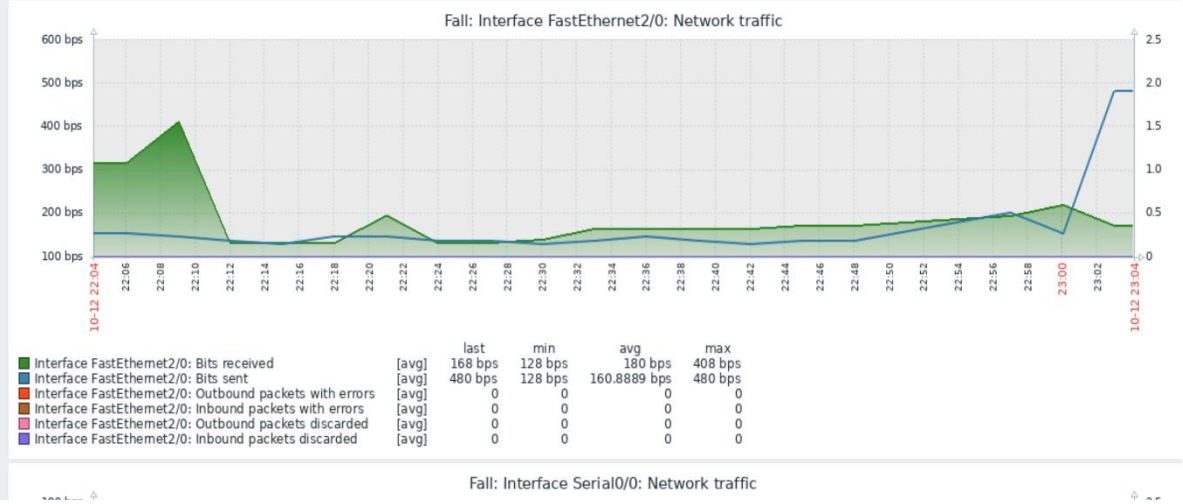


Figure A13: Network Traffic of the Interface FastEthernet 2/0 of the Fall device

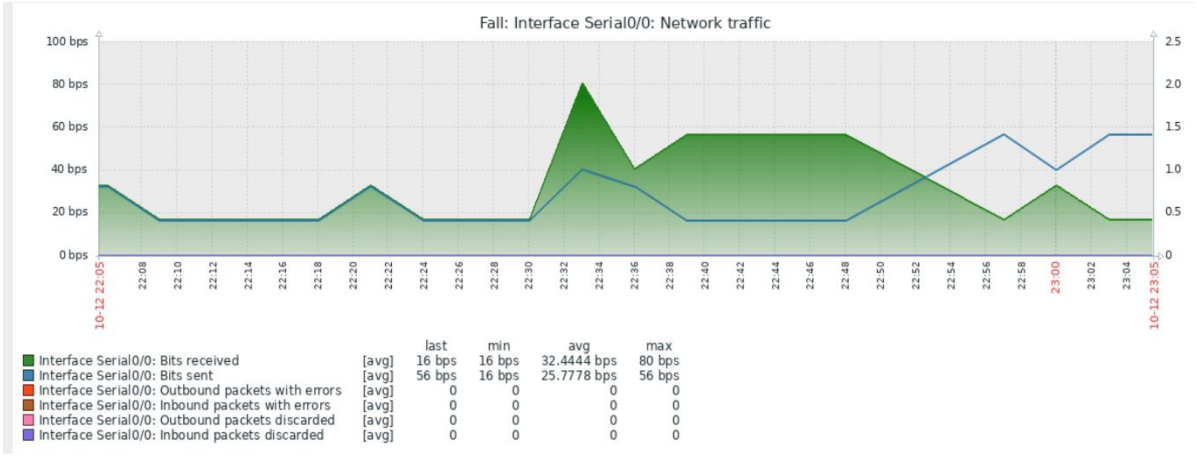


Figure A14: Network Traffic of the Interface Serial 0/0 of the Fall device

Figure A13, A14 and A15 demonstrate the graphical view of the network traffic of the device Fall. Zabbix monitoring tool has several advanced functionalities, such as creating an item that assists in applying alerts for the Zabbix server to monitor the network proactively and triggers the alert if the configured alert meets the alert condition. These are the advanced features we

have used to apply our critical checkpoint threshold and the SLA checkpoint threshold and monitor the network proactively.

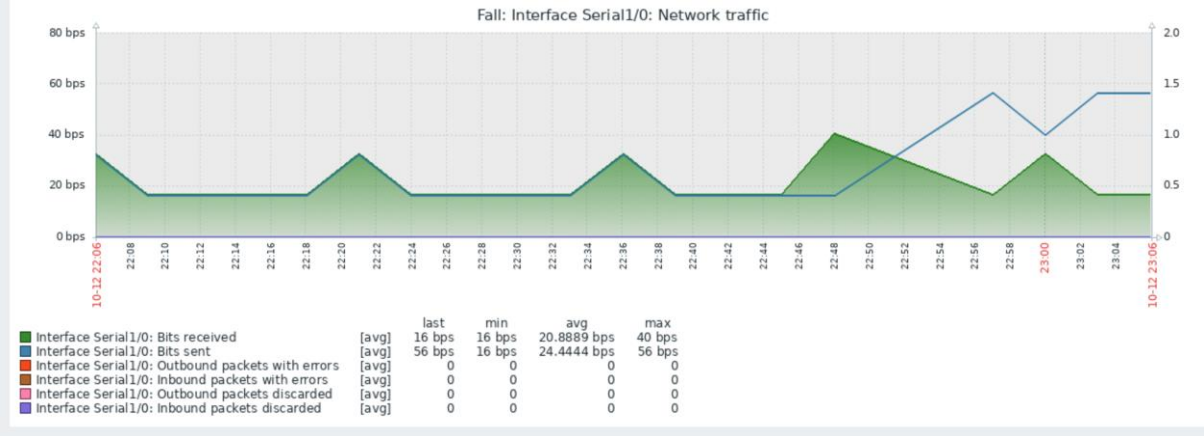


Figure A15: Network Traffic of the Interface Serial 1/0 of the Fall device

2. Passive monitoring

We have used Rapid Miner to identify the network traffic category that assists in the traffic classification process of our research. We have introduced an unsupervised machine learning-based support vector machine (SVM) approach to identify traffic. The details of the traffic identification process are discussed in chapter 6.

2.1 Traffic Identification:

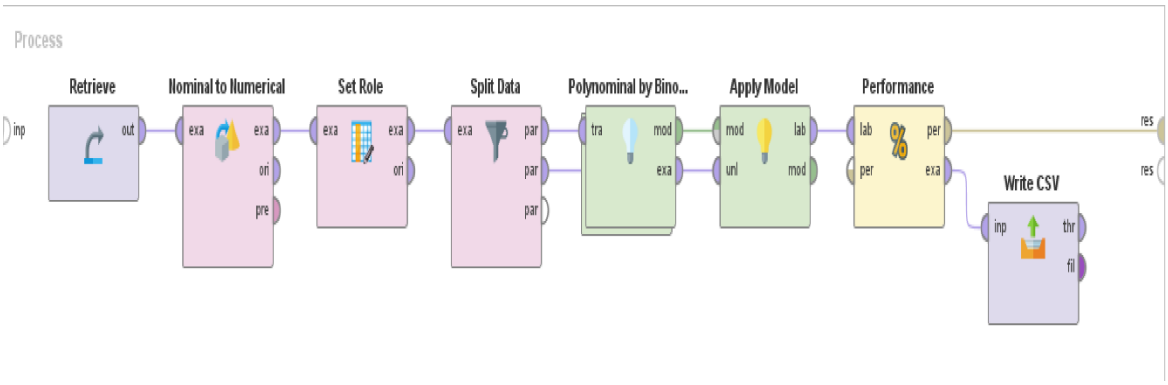
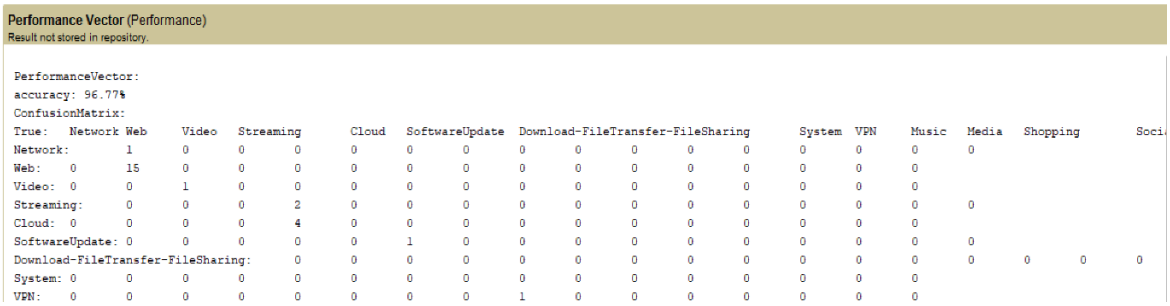


Figure A6: Design View of the Rapid Miner Operators for Traffic Identification

In order to implement the unsupervised machine learning (Support Vector Machine), The operators of the Rapid Minor are shown in figure A16. In this approach, we have used the network traffic data that is extracted from the proactive monitoring continuous monitoring stage. In this process, we retrieve the dataset from the retrieve operator and then convert the nominal data into numerical data as the SVM model can not operate with nominal data. Then we set the target role as Label and other related features. We split the data set in this stage into the training and testing dataset groups. The training dataset group we have used to train the model to determine the traffic category that assists the model in identifying the traffic category accurately.

We have used the last two operators to determine the results. The performance operator is used to determine the model's accuracy, and the accuracy is shown in figure A7 as 96.77%. In addition, the write CSV operator is used to get the results in a CSV format that we can use for other experiments.



Performance Vector (Performance)
Result not stored in repository.

```
PerformanceVector:
accuracy: 96.77%
ConfusionMatrix:
True:   Network Web     Video   Streaming       Cloud   SoftwareUpdate  Download-FileTransfer-FileSharing       System  VPN     Music   Media   Shopping        Soci
Network:        1       0       0       0       0       0       0       0       0       0       0       0       0       0       0
Web:    0       15      0       0       0       0       0       0       0       0       0       0       0       0       0
Video:  0       0       1       0       0       0       0       0       0       0       0       0       0       0       0
Streaming:      0       0       0       2       0       0       0       0       0       0       0       0       0       0       0
Cloud:  0       0       0       0       4       0       0       0       0       0       0       0       0       0       0
SoftwareUpdate: 0       0       0       0       0       1       0       0       0       0       0       0       0       0       0
Download-FileTransfer-FileSharing:      0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
System: 0       0       0       0       0       0       0       0       0       0       0       0       0       0       0
VPN:    0       0       0       0       0       0       0       0       1       0       0       0       0       0       0
```

Figure A7: Performance Vector accuracy results.

End of the Thesis

Appendix B

(SELECTED PUBLICATIONS)

Future Generation Computer Systems 119 (2021) 176–187

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Guaranteeing end-to-end QoS provisioning in SOA based SDN architecture: A survey and Open Issues

Shuraia Khan [a,*], Farookh Khadeer Hussain [a], Omar K. Hussain [b]

[a] School of Computer Science, University of Technology, Sydney, NSW, Australia
[b] School of Business, University of New South Wales, Canberra, ACT, Australia

ARTICLE INFO

Article history:
Received 19 November 2020
Received in revised form 9 February 2021
Accepted 15 February 2021
Available online 19 February 2021

Keywords:
Quality of Service (QoS)
Software dEfined Networking (SDN)
Service Oriented Architecture based SDN

ABSTRACT

Ensuring end-to-end Quality of Services (QoS) is a challenging aspect in traditional network architectures. Software-Defined Network (SDN), as the new norm of the network, has ascended in response to a traditional network's limitations. SDN's benefits are its ability to provide a global networking view, programmability, decouple the data plane with the control plane. Integrating SDN architecture with Service-Oriented Architecture (SOA) paradigm brings a novel network-based notion for service delivery. However, it also introduces new challenges for maintaining the QoS in these networks. Researchers from both academia and industry have proposed and developed several resolutions for QoS management in SDNs. However, gaps still exist in developing and applying such resolutions for QoS management in SOA-based SDNs. This review paper aims to identify these gaps by representing a sketch of the effectiveness of the existing QoS management techniques in SOA-based SDNs. We first identify the four different requirements that QoS management techniques need to meet to be applied in SOA-based SDNs. We then categorize the relevant QoS management approaches into five main categories of *QoS based controller design*, *Resource allocation-based approach*, *Queue scheduling and management-based approach*, *QoS-driven optimal routing*, and *Service Level Agreement (SLA) based quality management in SDN*. We then compare the working of techniques in each category against the identified requirements for guaranteeing end-to-end QoS provisioning in SOA based SDN architecture and present directions for future research.

1. Introduction

The success of the rapidly advancing computing technologies and applications such as web browsing, email, VoIP, audio and video conferencing, streaming, online gaming, texting, and e-commerce is highly dependent on the Internet. The importance of need and humankind's dependence on these technologies has been seen during the COVID-19 pandemic. These technologies have characteristic flows and demand a highly diverse and dynamic network to deliver on their requirements. From the viewpoint of service provisioning, this brings in different types of challenges, ranging from ensuring the security of communication to guaranteeing the Quality of Service (QoS) being delivered [1]. Our focus in this paper in on guaranteeing the QoS aspect during service provisioning. Thus, we focus on the network's capability to adapt to the need of the technology and application to deliver the expected service from it. Traditional networks are decentralized in nature and utilize a static architecture. This places numerous constraints on it to dynamically adapt to the need of the application. To address these limitations, Software-Defined Networking (SDN) technology was proposed. SDN is defined as a model that intelligently supports applications with dynamic needs while depressing operating costs by providing a simplified way to manage hardware and software [2]. As opposed to a traditional network, SDN decouples the data and control planes and centralizes the whole network's controlling process [2]. This enables the centralized control of the network, dynamic management, and optimization of the flows, resources, and infrastructure. As a result of these advantages, SDN has drawn significant attention from network service providers to make their networks' best use [2]. Researchers have also explored how SDN can be applied along with ensuring the QoS delivery in SDNs [3–5]. One such way, that is the focus of this paper is Service Oriented Architecture (SOA) based SDN.

As the principle of SOA is to offer a dynamic composition of services, SOA based SDN is a network that is service-based, loosely coupled, reusable, and can be abstracted [6]. In other words, SOA based SDN achieves the SOA principle in network management by incorporating several smaller services into specialized services [7]. Doing so makes the network architecture

* Corresponding author.
E-mail addresses: shuraia.khan@uts.edu.au (S. Khan), o.hussain@adfa.edu.au (O.K. Hussain).

https://doi.org/10.1016/j.future.2021.02.011
0167-739X/

flexible and adaptable to either scale up or down according to specific application needs in changing environmental conditions. Such an architecture has various advantages, especially in autonomous systems, where the required network capacity can be achieved to meet their diverse requirements at different periods. However, before such benefits can be achieved, various challenges need to be addressed to guarantee end-to-end QoS provisioning in SOA-based SDN architecture. Some of those challenges are (a) negotiating the network required as a service to achieve the expectations, (b) selecting a capable network provider that can deliver on the negotiated requirements, (c) ensuring that the network being provided meets the QoS requirements by managing and taking proactive decisions that will ensure the expected expectations are met. As shown in Fig. 1, these challenges as tasks are not present in the management of traditional SDNs. However, from the viewpoint of SOA-based SDN, they are essential to be addressed for the networking resources to be abstracted and utilized on-demand by users or autonomous systems in the form of a Network-as-a-Service (NaaS) paradigm [8]. To achieve this, the data and control plane of the SDN architecture requires another dimension of decoupling between service functions and network infrastructure to manage them dynamically [1].

2. The motivation of the paper

Existing literature has attempted to address these challenges by using Network Functions Virtualisation (NFV), which refers to cloud networking concepts. This technology has been applied in the telecommunications space by which network functions are deployed as virtualized software instances instead of dedicated hardware appliances. This enables creating a logically isolated network partitions over shared physical network infrastructure that allows the aggregation of multiple resources as single resources [9]. Researchers have combined SDN with NFV to deliver the network functionalities in a service-oriented way [9]. However, most of the work is at a theoretical stage. This leaves the objective of how to guarantee end-to-end QoS provisioning in SOA-based SDN as an open question.

In the networking context, QoS measures a network's capability in different parameters while delivering it as a service. Specific to computer networks, QoS can be measured in various parameters such as inadequate bandwidth, end-to-end traffic delay, jitter and packet loss [10]. Guaranteeing end-to-end QoS provisioning means ensuring that the delivered network meets the negotiated expectations. Cisco Internetwork Operating System (IOS) defines three types of architectures: best effort, integrated and differentiated, which provides a network with different level/s of guarantees. The best-effort service is also known as a lack of QoS model. This is because it gives no guarantees, and the network delivers data based on its capability, without any assurance of reliability, delay bounds, or throughput [11,12]. Best effort service is suitable for a wide range of applications where the data delivery assurance is not mandatory, such as general file transfer or emails [12].

Integrated service architecture commits to meet the defined QoS requirements of the application. In this architecture, the application requests a network of specific requirements before sending the data [12]. These architectures are suitable for video or sound applications and that need to be delivered without interruption. Differentiated service architecture provides a scalable network with differing level/s of QoS as required by the application [12]. Such architectures are mainly used for mission-critical applications. Depending on the necessary scalability level, a differentiated network can further be classified as either *hard* or *soft QoS* types. Hard QoS represents the "absolute reservation of network resources", as done in integrated service architecture [12]. Soft QoS reservation of network resources, on the other hand, are not firm. In such a case, the network provided may not meet the defined requirements for a period. All these types of architectures are available when autonomous applications use SOA based SDN to request network capability for their requirements. However, according to these architectures' characteristics, not all of them will suit the needs of an application. Thus, to guarantee end-to-end QoS provisioning in SOA-based SDN, a key requirement for the requesting applications is to follow a systematic process that will help them negotiate, select, monitor, and manage the right type of network with providers who can deliver on their specific requirements. If this process is not followed, then there is a very high chance of the delivered network QoS not meeting the requirements. As shown in Fig. 1, incorporating this process for guaranteeing the QoS of SOA based SDN requires a paradigm shift in how network as a service is formed and managed as compared to how it is done in a traditional SDN model.

The motivation of this review paper is to identify and discuss the existing QoS guaranteed approaches in SDN and analyse if they meet the specific requirements needed for forming informed networks between autonomous services in SOA based SDN. The key contributions to the literature arising from this review are as follows:

- *Identifies the key requirements to guarantee end-to-end QoS provisioning in SOA based SDN architecture:* We recognize the four different requirements that need to be measured to determine or analyse if the services being delivered in SOA-based SDN meets the defined QoS or not.
- *Identifies the shortcomings of existing QoS management approaches in SDN to be applied in SOA-based SDN architecture:* We categorize the existing QoS management approaches in SDNs into five types and study the working of each approach. We then determine if they can be applied in SOA-based SDNs to be guaranteed QoS in those applications.
- *Introduce future research directions:* Based on our comparative analysis, we then identify and provide potential research directions for QoS management in SOA-based SDNs.

The paper's remainder is organized as follows: Section 3 identifies the key requirements to guarantee end-to-end QoS provisioning in SOA-based SDN architecture. Sections 4–8 discuss the QoS management in SDN by using controller design-based, dynamic resource allocation approach, queue scheduling based, optimal routing approach, and SLA-based quality management approach respectively, and their limitations in guaranteeing end-to-end QoS guarantees in SOA-based SDN. Section 9 summarizes the challenges and outlines the directions of future research. Section 10 concludes the paper.

3. Key requirements to guarantee end-to-end QoS provisioning in SOA based SDN architecture

To identify the key requirements in guaranteeing end-to-end QoS provisioning in SOA based SDN, we take inspiration from Cloud Computing, one of the successful computing architectures that have arisen from SOA. Cloud Computing offers computing and storage resources on-demand as a service to its users in its simplest form. This enables the users to use these resources on a PAYG model rather than spending the vast costs upfront in acquiring them physically. Our focus in this paper is on how cloud services between users and providers are formed. In this process, researchers have identified many different criteria that range from certifications to migration support, which must be considered when deciding on a service [13]. One such criterion, which is our focus in this paper, is forming Service Level Agreements (SLAs) with reliable and reliable service providers who can deliver on the needed expectations.

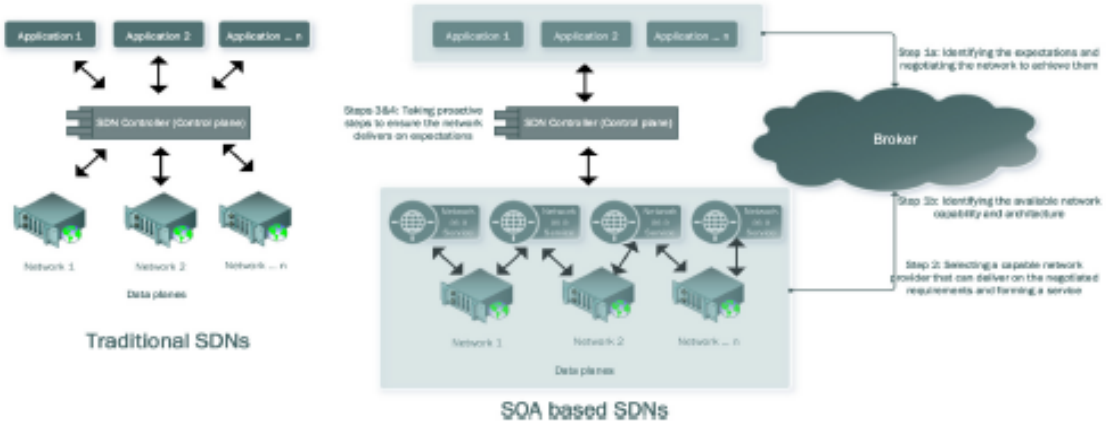


Fig. 1. Difference between a traditional and SOA-based SDNs.

Service Level Agreements (SLAs) are agreed between a service provider and a service consumer [14]. The intention of the managing service level agreement (SLA) is to ensure that the defined services meet a certain level of criteria that have been demarcated in the agreement. The SLA also contains the service's non-functional requirements termed as the Quality of service (QoS) terms. These terms include the obligations to be met, pricing for the services offered and penalties if the agreement is not achieved [15]. Failure to meet the agreement's conditions may result in SLA degradation or SLA violation [16]. Service degradation or service violation may create a significant impact or loss on the total business outcome. Research in this criterion comes under the broad area of Cloud service selection, which ranges from service discovery [17], service negotiation and selection [18], proactive service management to prevent SLA violations [19], service migration, if needed [20] and lastly ranking service providers according to their performance [21]. The literature mentions that having such measures is beneficial both to the service users and the service providers. From the service provider's perspective, it enables the creation of healthy competition among them [21]. The service user's perspective enables them to make informed and smart service-based decisions that will enable them to achieve their expectations in a decentralized environment [22]. In other words, the Cloud service selection process does not follow a one-size-fits-all approach for all the service selection decisions, but it follows a customized process that first identifies the available services, negotiates between the service providers and the service users to form expectations, forms an SLA based on those expectations, and then proactively manages them to ensure that the QoS of the service delivered matches these expectations, before ranking the service provider on a scale. Similarly, it is our purview that if a SOA based SDN needs to deliver a network that satisfies the requirements of the different services, it must follow a similar approach that would first ensure the correct identification of services according to the applications needs, forming SLAs with them, and then managing them to ensure that the expectations are achieved. To achieve such an approach to guarantee end-to-end QoS provisioning in SOA based SDN architecture, the requirements to be met are:

3.1. Ability to personalize a service's required QoS (hereafter considered as R1)

Personalization helps in meeting an application's (or consumer's) demands more effectively and efficiently, where the 'one-size-fits-all' approach is not applicable. In SOA-based services, personalization is a feature that can be used during service selection and which is beneficial to both the application users and network service providers. To the application users, it guarantees that the network to be delivered will meet their requirements and that they will pay only for the network they use. To the network service providers, it assists in knowing the QoS to be delivered and deciding if they have the required capability to form an SLA with the specific application. Personalization is also termed as negotiation after which the service requirements are demarcated in the SLA [23]. While the benefits of personalization or negotiation have been applied in various domains such as Cloud Computing, it has not received the same type of exposure to personalized network services in SDN. As a result, there are limited opportunities to receive personalized QoS delivery in SDN based on an application's needs, which is one requirement to deliver and use resources by using an SOA-based network.

3.2. Reputation value-based network service selection (hereafter considered as R2)

In the consumer market, a service or product's reputation value is widely used to make a fair judgement on it. This is especially beneficial when there are a wide variety of possible options available, and the consumer wants to select the one that best matches their needs. A provider's higher reputation value represents a higher level of service satisfaction from the users, which brings it more trust among the potential consumers. Offering reputation-based network service selection in SDNs will enable the requesting applications to decide which service provider to choose based on the QoS requirements. These need to be considered when the network is being delivered as a service, based on the provider's reputation who is committing to deliver what it has promised.

3.3. Measuring the satisfaction of a network service provider based on the QoS it provides (hereafter considered as R3)

To achieve requirement 2, one key aspect needed is the ability to measure service providers' satisfaction rating. This should be determined by comparing the QoS values that the SDN service providers are committed to provide with their services defined in the SLAs (also termed as QoS guarantees) against the actual QoS values deliver (also termed as *QoS actual delivery*). Furthermore, the determined satisfaction rating should not be subjective and ambiguous and should be determined on a globally agreed scale by a commonly agreed process [24]. This is important as the SOA-based SDNs can be requested as a service by different geographically diverse users. Hence, if each application determines

Table 1
Role of R1–R4 in end-to-end guarantee of QoS provisioning in SOA-based SDN.

	R1	R2	R3	R4
Service customization	√	×	×	×
Service selection	×	√	×	×
Service discovery	×	×	√	√
Service quality	×	×	×	√

the satisfaction value by using its own way, it does not represent a true representation of a service provider's ability to deliver on the promised QoS. The existing literature in SOA-based SDN does not have a mechanism to measure and quantitatively represent the network service provider's satisfaction on a standardized scale or have a standardized process. Without this, the service provider's satisfaction which represents an important criterion during the network service selection phase, cannot be considered.

3.4. Measuring the network provider's QoS actual delivery by considering its composite nature (hereafter considered as R4)

Given the flexibility of SDNs, providers offer guarantees of services in different aspects, such as network reliability, scalability, performance under latency limitations which come under categories such as application-based, and network-based. All these guarantees are termed under QoS guarantees, that have been explored by several scholars in the different aspects [2,25]. However, to facilitate requirements 2 and 3, the QoS actual delivery values of each service provider should be represented in each individual aspect, and not just as a composite measure. This is important to differentiate the network providers in terms of their QoS. This concept has been utilized in cloud service selection where methods such as multi-criteria decision making (MCDM) are used by service users to rank the service providers according to their specific requirements and then choose the best possible one [21,26–28]. To facilitate and ensure that SOA based SDNs are used according to their potential, a composite representation of QoS provided by each network provider along with the QoS measure provided in each aspect is needed to be represented for each SDN. Furthermore, to ensure that SOA-based SDNs can be requested as a service by different geographically diverse users, such QoS guarantee values should be maintained in a QoS repository that is globally accessible.

The combination of these four requirements (R1–R4) will assist towards achieving end-to-end guarantee of QoS provisioning in SOA-based SDN architecture. It does that by assisting service users and service providers by forming informed SLAs by following a systematic series of steps which start from service customization and end by service quality measurement. As shown in Table 1, R1 assists in the task of service customization. Using this task, the service users can personalize the service needed from a provider according to their specific requirements. R2 assists service users in the task of service selection. Using this task, service users can utilize the determined reputation value of service providers in their service selection decisions. R3 and R4 assists the service users can provide their feedback about the Quality of the service which they received from a provider in different criteria. This will assist other service users in the tasks of service discovery and determining service quality in the different QoS metrics. As the combination of these criteria assists in the end-to-end guarantee of QoS provisioning in SOA-based SDN, these four requirements form the basis of our investigation into the existing approaches utilized for guaranteeing QoS in SDNs. Our objective is to identify if the existing QoS management approach proposes a solution to achieve these requirements that will facilitate SDN to be delivered as a service in an SOA based model. We categorize the existing approaches to achieve QoS in SDNs into five groups, namely *controller design-based QoS management*, *dynamic resource allocation approach for QoS guarantees*, *queue scheduling based QoS management*, *optimal routing approach for QoS guarantees*, and *SLA-based quality management*. In the next sections, we introduce each of the above-mentioned approaches for QoS management to compare if they address the requirements needed for guaranteeing QoS in SOA based SDNs.

4. Controller design based QoS management

Controller design based QoS management enables network managers (controllers) to centrally configure and manage the network. An SDN's control plane encompasses of one or more software regulated SDN Controller(s) that enable it to control and manage network functionality by administering the traffic forwarding behaviour through Controller-Data Plane Interface (C-DPI). Functional Components and Control Logic are the two foremost elements of SDN Controller where the functional component manages the control behaviour and control logic component and maps between networking requirements for application demand and network element resources accordingly [2]. As the controllers can obtain the global view of the network, they use this to apply control policies to the network to manage the SLAs. Researchers have used the specifics of the architecture and modified it to propose new designs for guaranteeing QoS in SDN. The core concept of this approach is modifying the control logic component, implement policies based on SLA specifications or application or user demands. The functional component brings out the controller behaviour as well as network behaviour based on the response of control logic components.

To achieve QoS in SDN based cloud infrastructure, Govindarajan et al. [29] proposed a controller reference architecture named Q-Ctrl. Q-Ctrl conjoins with another QoS lifecycle model to achieve effective QoS operations in SDN. It aims to achieve end-to-end QoS for enterprises, multimedia, scientific applications and execute them in a virtual overlay network via Open vSwitch (OVS) to work in either direction, controller, or simulator mode. In this approach, QoS requests are scheduled to QoS flow injector that ensures that the achievement of the QoS in an open flow enabled networking devices. The QoS lifecycle proposed by researchers is classified into five major operation tasks such as Queue creations, QoS flow addition, QoS flow modification, QoS flow deletion, and Queue deletion. Q-Ctrl has been tested on Web and Video Streaming applications and the results depicted that the allocation of bandwidth has been regulated effectively. Controller Placement Problem (CPP) is one of the major challenges in achieving QoS in SDN [30]. Based on the locations of switches, CPP is defined as choosing suitable locations for "K" controllers so that the latency between the controllers is minimized while maximizing propagation delay, network reliability, load distribution, and failure resilience. Cheng et al. [31] studied a QoS guaranteed CPP solution where the focus was on placing the required controllers that can minimize the response time between them. Therefore, two heuristic algorithms, namely the primal–dual-based and network partition-based algorithms were proposed and tested on the Internet Topology Zoo database (database includes publicly available network topologies) in a simulation environment. The result shows that the Incremental Greedy Algorithm slightly outperforms compared the other two at the sacrifice of response time.

Egilmez et al. [32] propose a novel open flow controller design based framework named OpenQoS for an end to end QoS services on multimedia delivery over SDN. The proposed approach groups the flows of multimedia and incoming traffic by dynamically placing the multimedia flows in QoS guaranteed routes

and the data flows in the traditional shortest path route. The proposed framework develops a QoS architecture and then runs the OpenQoS framework on it. It also develops a dynamic QoS routing based optimization framework to manage multimedia traffic. To measure the performance, the researchers conducted experiments over a real SDN test environment and compared the performance results with the current state-of-the-art, HTTP-based multibit rate-adaptive streaming. The experimental results represent that OpenQoS based approach guarantees in users experiencing seamless video delivery. SDN protocols such as OpenFlow enable traffic control on a per-flow level which is one of the pre-requisites for an end-to-end performance guarantee in SDN. Tomovic et al. [10] propose an SDN-based control framework for QoS provisioning that mainly presents on the original design of the SDN/Open Flow controller environment and provides bandwidth guarantee for priority flows in an automated manner. This control framework enables automated and flexible control on network devices by programming the controller to achieve the required QoS level for multimedia applications. In terms of QoS provisioning, several service models and mechanisms, such as IntServ/RSVP, Diffserv, and MPLS are proposed. They also execute centralized control monitoring and determine an overall representation of the network's resources for smart traffic management. In addition, they also minimize the degradation of best-effort traffic by providing priority flows. To compare their proposed solution with traditional techniques they utilize the best-effort service model and Int-Service and demonstrate that their model outperforms the best-effort shortest path routing and IntServ. Adami et al. [33] propose an approach where the network control application is built on top of the existing Floodlight tool for end-to-end QoS provisioning in SDN. The authors modify the Dijkstra algorithm to provide an efficient routing according to the traffic load. By considering the network status as QoS metrics and ensuring no excessive devices or links are attached that may result in slow transmission and packet loss, they develop a system to guarantee QoS for some specific traffic groups. The experiment on the emulation environment demonstrates the system to be behaving as expected and managing network resources efficiently thereby providing guarantees on traffic handling.

To summarize, the controller design approach is one of the popular approaches for SDN management which has been utilized in different applications that range from providing a QoS based solution to the entire SDN network to providing application-based solutions such as Multimedia applications. However, as shown in Table 2, the controller design approach has drawbacks which do not commit to the requirements of SOA based SDN for them to provide end-to-end QoS guarantee in it.

5. Dynamic resource allocation approach for QoS guarantees

Resource allocation in the computer networking context is a process of assigning and managing the available network assets or resources in an optimized manner to support the network's demand. The resource allocation approach is also known as dynamic resource allocation for priority users or resource reservation approach. QoS guarantee resource allocation provides the corresponding network services for some applications to satisfy users' demands such as performance metrics, latency and bandwidth. To achieve QoS in SDN networks, researchers have used resource allocation approaches in several contexts especially in multimedia applications. SDN through resource allocation and scheduling mechanisms enables the QoS guarantee [34]. Using such functionality, researchers have developed an innovative and incremental framework for dynamic network management as well as allocating resources named SoIP. The proposed model enhances the capability of the QoS guarantee on the Internet by building a software-defined overlay network over the IP network. This technique brings the advantage of per-flow management characteristics in SDN to meet the aligned applications demand of resources. This framework also represents that a seamless combination of SDN and IP networks is achieved by the proposed resource scheduling mechanisms in order to guarantee QoS. The proposed model has been tested in an emulation environment in which the results demonstrate SoIP's ability to meet end user's QoS requirements in terms of the defined performance metrics.

Failure recovery is one of the important challenges which service providers face while achieving QoS in a network. The researchers propose a framework to guarantee QoS by using a vendor-agnostic interface of SDN technologies such as OpenFlow, OF-Config, and OVSDB [35]. Between best-effort and high priority traffic, the working logic of the proposed approach is to prefer the latter before the former. The framework has been tested on a single Autonomous System (AS) (emulated pan-European topology) and multiple AS Scenarios (designed in the City Flow project on the OFELIA testbed facility in iMinds) with valuation among three failure recovery scenarios. This approach reserves resources for the high-priority flow of entrance switch whereas Open QoS does not use any resource reservation scheme. The mechanism of this framework consists of several following steps that start with configuring three default queues on each port using the OVSDB protocol. The controller then runs OSPF, BGP protocols on each router using Route Flow followed by establishing flow entries in the router. The aim is to configure and ensure the availability of network resources for high priority traffic with rate limiter queue. Using such an architecture, even in failure conditions precedence to high priority traffic was given to best-effort traffic. Open Flow networks/SDN provide bandwidth guarantee by using well-known FIFO scheduling. As a result, OpenFlow switches might not meet the QoS requirements for some applications such as Multimedia streaming. This limitation creates a packet scheduling issue. Airton et al. [36] propose a QoS development strategy (QoSFlow) for OpenFlow to enhance controlling multiple packet schedulers of the Linux kernel. The framework contains a QoS module that has been added to the standard OpenFlow data path. The QoS module used consists of three components, namely traffic shaping, packet schedulers, and enqueuing. OpenFlow 1.0 with FIFO scheduler has been used to achieve different treatments for the packet. The strategy has been tested in a test environment where an Open WrtBackFire has been installed according to Pantou projects and the experiment has been performed based on several parameters such as response time, switch capacity, number of queue's impact, bandwidth isolation, and QoE evaluation. The evaluation result showed that the proposed approach has a low response time and the use of Stochastic Fairness Queuing (SFQ) brings improvements (an increase of more than 48% on PSNR value) on QoE. Akella et al. [25] propose an approach to resources of the network using cloud carriers upon users' requests and subject to QoS requirements. Jeong et al. [37] propose a QoS aware Network Operating System for SDN service provisioning using generalized Open flows. Kassler et al. [38] propose a system to enable negotiation among services, network communications between end-users and allocate network paths for sending multimedia flows according to the service configuration as promised. This system has the capability of Quality of Experience (QoE)-driven path allocation and optimization for multimedia services in SDN. A multi-tenancy management based framework is proposed by Alba et al. [39] to ensure Quality of Services in SDN through tenant isolation, prioritization, and flow allocation. This management framework proposes a virtualization technique (Virtualization-enabled Routing (QVR)) that can be implemented in SDN architecture and combine this technique with the proposed QoS aware framework

Table 2
Controller Design or Controller placement Approach for QoS Guarantee of Software Defined Networking.

Study	Algorithms/Architecture	Contributions	Area of applications	Implementation	R1	R2	R3	R4
Govindarajan et al. [29]	Q-Ctrl model with QoS lifecycle (Bandwidth Allocation)	Effective	Web and Video Streaming	Simulation & Real Network	×	×	×	×
Cheng et al. [31]	Incremental greedy, Primal–dual based, Network partition-based	–Incremental Greedy Outperforms, –Network partition less response time	Global	Simulation	×	×	×	×
Egilmez et al. [32]	OpenQoS framework, Dynamic QoS routing	Guarantee seamless video delivery	Multimedia	Real Network/ Test Bed	×	×	×	×
Tomovic et al. [10]	Per-flow control, automatic priority flows	–Automated Bandwidth guarantee for priority flows	Multimedia, Other applications, monitoring, control of resources	Real Network/ Test Bed	×	×	×	×
Adami et al. [33]	Modified Dijkstra Algorithm	–More Efficient Resource management, –Guaranteed traffic handling	Global	Simulation	×	×	×	×

to allow flow allocation with respect to different tenant applications. In another work, Karaman et al. [40] analysed a high-quality uninterrupted VoIP service with a video to be used by users using the SDN. The experiments of this study are conducted on an output of the conjoint works of OFERTIE and SIGMONA projects. The results emphasized on the QoS improvement that reduced the loss experienced to lower than 5% as well as reducing the latency and jitter by more than 50%.

A framework for applying Service-Oriented Principle (NaaS) in SDN is proposed by Duan [8] to address the challenging issue of end-to-end QoS provisioning. The key functions of end-to-end QoS provisioning are the allocation of sufficient bandwidth in network services. A NaaS in the SDN framework sits on top of an SDN controller or a Set of SDN controllers. A NaaS abstraction Interface is between the SDN Controller domain and the orchestration module to abstract the network functionalities of each domain and is named as Network Service. The orchestration module takes the service requests from the upper-layer applications module and then determines the required amount of bandwidth for each involved network service and then acquires the bandwidth to achieve given end-to-end QoS requirements. The "Arrival Curve concept" in network calculus is employed in this framework for developing a general abstract load profile. Analytical and Numerical results show that this framework is able to support end-to-end QoS with flexible service management and higher bandwidth utilization. To summarize, the allocation of network resources dynamically is one of the effective approaches to achieve QoS delivery in SDN Network. A comparative analysis of the techniques proposed to achieve this aim is shown in Table 3. This comparative analysis clearly identifies that such approaches are preferable for multimedia applications where the availability of network resources is crucial. However, these approaches do not guarantee the provision of QoS delivery in a global network. Furthermore, these approaches are unable to provide personalized service delivery for the end-user. Most of the techniques have been evaluated in a simulation environment; as a result, there are fewer opportunities to understand how the techniques will behave in a real network as well as in the Service Oriented Architecture based SDN environment.

6. Queue scheduling based QoS management

In a network, the packets are processed in a queue that follows the First in First out (FIFO) queue processing rules. However, some packets in the queues require higher priority to process than other packets which are ahead of them. This queuing technique makes a significant impact on QoS service delivery along with traffic shaping [2]. In addition, multiple packet schedulers require a more optimal queuing technique to maintain the required network performance. Queue scheduling technique is a way of mapping a packet to an internal forwarding queue based on QoS requirement information and driving the queues according to a predefined queuing policy.

Li et al. [41] propose a queue scheduling based QoS techniques for cloud applications in SDN. The approach identifies applications and determines the required QoS levels for each application type. It then implements a queue scheduling technique to permit delay-sensitive data to be de-queued and sent first. In the system design stage, there are three main modules to control and manage messages. Control Message Module forwards the messages according to flow table rules whereas Queue Management Module configures queues based on configuration information. Queue Scheduling Module schedules the packet out of the queues with different priorities. The evaluation of this approach has been conducted through both experimental and theoretical analysis. Theoretical analysis shows that this method can provide differentiated services for application flows and is able to map to different QoS levels.

Cosmin and Jose [42] propose an Application Programming Interface (API) named "QoS Config API" for configuring QoS resources dynamically by allowing applications to configure priority queues to the ports of network devices. To implement the proposed API, the OVSDB protocol has been used at D-CPI (Enables programmability of the entire network) of an existing SDN Controller (SDNC). The implementation has been conducted on a distributed testbed. The first test demonstrates that QoS Config in the data plane by using rate limiting queue can dynamically control and allocate bandwidth to different traffic flows. Moreover, there is a possibility of creating high-level abstractions from APIs exposed by the SDNC. The second test demonstrates that by the current level by which QoS Config API works, it is able to dynamically configure of QoS. Sharma et al. [43] developed a QoS framework for OpenFlow by configuring the business consumer traffic to a high priority queue in all conditions (e.g. Network Failure condition). This approach divides the traffic into business and best-effort traffic followed by configuring priority queue and then redirects different flows to suitable priority queues. Open Flow protocol conjoining with Open vSwitch Database Management (OVSDB) configuration protocol has been used to provide high QoS. The experiments have been performed in an emulated Openflow environment named OFELIA testbed facility that is provided by iMinds. It was observed that upon experiencing failure, this framework can re-establish flow entries on the affected paths and the edge router is able to reconfigure its rate limiter queues

Table 3
Comparative analysis of dynamic resource allocation-based approaches for QoS guaranteed service delivery in SDN.

Study	Algorithms/Architecture	Contributions	Area of applications	Implementation	R1	R2	R3	R4
Hu et al. [34]	"SoIP " framework (resource allocation)	Potential	QoS Guaranteed Internet	Simulation & Real Network	×	×	×	×
Sharma et al. [35]	Open-Flow, OF-Config, OVSDB	–Suitable (Future Internet) –Priority traffic get precedence	Global	Simulation	×	×	×	×
Ishimori et al. [36]	OpenQoS framework with FIFO schedulers	–Low response time –48% on QoE improvement	Multimedia	Test Bed	×	×	×	√
Jeong et al. [37]	Per-flow control, automatic priority flows	–Automated Bandwidth guarantee for priority flows	Multimedia & Other	Simulation	×	×	×	×
Poxcas et al. [39]	Virtualization-enabled Routing (QVR)	–More Efficient Resource allocation	Global	Simulation	×	×	×	×
Karaman et al. [40]	Resource allocation	Loss reduced	VoIP	Test Bed	×	×	×	√

appropriately along with the alternative available path. Durner et al. [44] determines the effect on network traffic when applying dynamic QoS mechanisms in OpenFlow-enabled SDN switches. The measurement study is accompanied by two fundamentally different Quality of service techniques named Priority Queue and Bandwidth Guaranteeing Queue. The result reveals a noticeable variation for different OpenFlow Switches in terms of performance. Additionally, different Queue implementation i.e. FIFO queue or SFQ queue, significantly impact on network performance. Table 4 shows a comparative analysis of Queue scheduling approaches with respect to the requirements for guaranteeing end-to-end QoS provisioning in SOA-based SDN architecture.

7. Optimal routing approach for QoS guarantees

Finding the best route while at the same time guaranteeing QoS for flows is one of the critical aspects for a network to meet. Calculating the most efficient route without delay is not easy since network resources can dynamically change anytime. In addition, due to the inability of traditional networks in having a global view of the network, per-hop decision making, and limitation in applying flow-based QoS, routing is an ongoing issue in them [2]. Moreover, propagation of high resource-demanding applications such as video conferencing and VoIP etc. on the Internet requires a more sophisticated, dynamic, and efficient routing mechanism that is able to meet the QoS demand. QoS driven routing is able to provide routing strategies that are capable of identifying the best paths to satisfy the maximum possible number of flows with QoS requirements. As a result, QoS driven routing is needed to keep flows under QoS guaranteed routes and provide optimized QoS.

A SDN based QoS control model for fast routing in an industrial QoS network is proposed by Guck et al. [45]. This model avoids control loop over forwarding and control planes; however, can route flows through multiple queue links with different QoS level, thereby maintaining QoS through Delay Constrained Least-Cost (DCLC) routing. The researchers developed an online algorithm for admission-control based on the existing DCLC routing algorithm. The result shows that this model-based approach can make accurate routing and admission control decisions within a few microseconds. Chen et al. [46] proposes a QoS guaranteed systemic approach for dynamic service chaining in SDN. Service chaining is the deployment procedure of building a sequence of individual service functions to accomplish a complex task. The authors developed an analytical method using Network Calculus (NC) and queuing theory to determine the delay performance and characterize them. The experiments have been conducted both on real network and virtualization networks using a Mininet simulation environment which demonstrates that NC delay analysis can provide deterministic QoS-guaranteed service chaining for any satisfied delay requirements. This approach is also able to design a network for future delay-sensitive Internet in which one of the factors to be guaranteed is the deterministic latency. Alharbi and Fei [47] proposes a QoS based framework for Smart Grid using Software Defined Networking that enables the allocation of Critical Flows in a dynamic manner. The authors divide the data traffic into two categories, namely *Best effort traffic* and *Critical traffic*. Best effort traffic does not have QoS requirements while the critical traffic has. The Control programme monitors the status of the network and redirects the critical flows over a better path by installing Open Flow rules in the switch. This is done by a path searching algorithm for the QoS path which is implemented as a module in the floodlight controller. For testing the QoS routing algorithm, a Mininet emulation environment has been used. The result demonstrates that, in comparison with the shortest-path routing algorithm, this approach improves the overall performance obtained by critical flows with the network achieving a higher level of utilization. Hongyu et al. [48] proposes a QoS guaranteed SDN based routing strategy. The proposed approach is aimed at the energy saving of the backbone network. The proposed technique integrates Backbone Networks energy-saving Strategy (BNESS) with Open Shorted Path First (OSPF) protocol with the Maximum Clique Problem (MCP) to search idle links to save energy. The approach has been tested using a Mininet simulation environment and the result shows that the BNESS algorithm is simple and can save energy effectively.

Akella and Xiong [25] proposed an approach for all priority cloud users in SDN by allocating bandwidth that can fulfil the demand for QoS requirements. This approach presents an efficient QoS routing algorithm by considering the congestion level, available bandwidth and hops count. A mathematical expression has been developed for path selection in a multi-cloud user environment to meet their specific requirements. It achieves this by (a) developing a metric that considers the available bandwidth, path length or hops count, and RTT in its analysis (b) and a queuing technique that can cater for multiple users in the cloud environment. A greedy algorithm has been used for path selection. This proposed algorithm can automatically switch paths that are available for higher priority clients to ensure the guaranteed QoS. The Utah Emulab testbed has been used to evaluate the performance of the proposed QoS guaranteed routing approach and the results represent the effectiveness of the approach. Seliuchenko et al. [49] propose a Multi-commodity flow allocation-based model in order to provide optimal routing based on QoS criteria. The proposed routing model allows balancing the

Table 4
Comparative analysis of existing queue scheduling approaches for QoS guaranteed software-defined networking.

Study	Algorithms/Architecture	Contributions	Area of applications	Implementation	R1	R2	R3	R4
Li et al. [41]	Application Identification, Queue Scheduling	Differentiated services.	Cloud Applications	Real Network	×	×	×	√
Caba and Soler. [42]	Rate limiting Queue	Dynamic QoS Configuration, Fine Granular service	Global	Distributed Test Bed	×	×	×	×
Sharma et al. [35]	Open Flow and OVSDB	–Restabilised the Open Flow after failure	Global	Test Bed	×	×	×	×
Durner et al. [44]	Dynamic QoS Mechanism	Different Queue significantly impact on network performance	Global	Real Network/ Test Bed	×	×	×	×

load based on the minimum-maximum load of network channels and the service quality of each stream. A flow identification technique had also been proposed to find the optimal set of routes through the network for all flows, with minimal total cost. The optimization finds the shortest path for high priority flows and distributes the low priority flows to align the network load. Table 5 shows a comparative analysis of QoS guaranteeing approaches with respect to the requirements to guaranteeing an end-to-end QoS provisioning in SOA-based SDN architecture.

8. SLA-based quality management

A Policy Based Management (PBM) approach has been proposed for managing the Quality of Services in a Software Defined Network [50]. This approach introduces an automatic policy refinement technique that can translate the SLA terms and conditions into a set of corresponding low-level rules. The high-level policies (SLA) are translated manually into technical requirements (SLS) and then translated into possible objectives (SLO). Through querying the QoS Class in an LDAP repository, the expectation to be achieved from each objective is determined. LDAP also contains a protocol list. e.g. the administrator sets the protocol H.323 in platinum class registered in the LDAP repository. The controllers receive the rules from the repository that is filtered by classes. Then the controller analyses the network flow and finds the best path to enforce the rules. As the best links between switches have been pre-populated, the proposed approach was shown to successfully reconfigure specific rules that improve performance. Bhattacharya and Das [51] propose a Dynamic Service Level Agreement that can provide QoS enabled network architecture to address QoS issues across the Internet. This proposed architecture offers interaction among internet service providers that helps to build dynamic relations to bring flexibility and adaptability. A single network consists of four NP (Network Providers) and 1 SP (Service Provider) domain where a source (SRC) and a destination (DST) relate to SP. A periodical database is stored by SP. When SRC wants to have a session with DST, SRC sends a request to SP. SP of source asks SP of destination to find the current destination network. By querying the database, SP knows all NPs involved in reaching the destination. NP also offers the best available path with QoS requirements. The implementation of a simple scenario and evaluation shows that the architecture is successfully able to find the best path before link breaks and able to find the best alternative path after link breaks.

A real-life implementation based technique in Software-Defined network is proposed by Körner et al. [52] that enable the application of QoS requirements using a conjoin approach of WS-Agreement standard and Open flow standard. The above capabilities can be achieved by implementing the following four subtasks: (a) creating Service Description Terms (SDT) Agreement that describes a part or complete service information therefore involving parties to understand the content of the SDT (b) developing an SLA framework named WSAG4J that implements WS-agreement and WS-agreement negotiation as protocol (c) Open Flow protocol defines the flows in the network and takes the forwarding decision and manages the Queues, and (d) Floodlight controller is an open-source controller that helps the module to discover the topology or forwarding path calculation. In addition, a cloud-based middleware has been implemented to apply negotiation using OpenFlow agreed quality constraints to the underlying cloud SDN substrate. The experiment result depicts that this framework can manage the network in a much easier way and calculate the overall network utilization.

From the above SLA based QoS studies, [50,51] are mainly using the Optimal Routing technique to find the best path for network flow after receiving the service requirements through Service Level Agreement. Both studies neither have an opportunity to select personalized services nor have measures to receive a guaranteed level of QoS. In the approach discussed by Körner et al. in [52], there is an opportunity to choose services and an ability to specify the QoS requirements and QoS level in the service negotiation stage. However, there are no measures defined for understanding QoS guarantees. The authors identify the broad range of Infrastructure-as-a-service (IaaS) offers to outsource the IT infrastructure into the cloud, whereas the properties of the underlying SDN network and its connected servers are treated as an infrastructure resource. There is no architecture detail or implementation detail available to achieve this Service Orientation structure in SDN. Table 6 shows a comparative analysis of SLA-based quality management with respect to the requirements for guaranteeing end-to-end QoS provisioning in SOA-based SDN architecture.

9. Guaranteeing end-to-end QoS provisioning in SOA based SDN architecture: Open gaps

Based on the above discussion, the Quality-of-Service delivery in SDN is a prime research challenge for SDN researchers and developers. This challenge leads to a big concern for SDN providers to ensure the user's satisfaction. While existing approaches have attempted to address this issue, the SLA-driven approach and most of the queue scheduling approaches have provided QoS solution for SDN networks, rather than an application or area-specific solutions. Moreover, very little research has been done to affirm the successful response rate after implementing their approaches. Most of the research has been tested in simulation, emulation environment or in Test Bed environment and not in the real SDN network. Furthermore, based on the comparative analysis, there are no approaches that can provide reliable personalized service delivery for service selection decision making. This, therefore, has issues in delivering reliable personalized services to the SDN consumer.

Table 5
Comparative analysis on QoS driven routing approach for QoS guaranteed software defined network.

Study	Algorithms/Architecture	Contributions	Area of applications	Implementation	R1	R2	R3	R4
Guck et al. [45]	–Online Algorithm on DCLC routing	–Accurate Routing –Admission Control Decision in Microsecond	Industrial	Simulation	×	×	✓	✓
Chen et al. [46]	–Network Calculus and Queuing Theory –Path Determination Algorithm	–Deterministic QoS Guarantee	Global	Real Network and Emulated Environment	×	×	×	×
Alharbi and Fei. [47]	Path Searching Algorithm for QoS Path	–Performance Improved, –Higher level Unitization	Smart Grid	Emulation	×	×	×	×
Hongyu et al. [48]	–OSPF with Maximum Clique Problem	–Simple and can save energy	Backbone Network	Simulation	×	×	×	×
Akella and Xiong [25]	–Matric based path selection –Greedy Algorithm –Queuing Technique	Effective	Priority Cloud Users	Emulab Test Bed	×	×	×	×
Seliuchenko et al. [49]	–Flow Identification technique	Delay is below the Threshold	Global	Emulation	×	×	×	✓

Table 6
Comparative analysis on QoS driven service level agreement management-based approach for QoS guaranteed software defined network.

Study	Algorithms/Architecture	Contributions	Area of applications	Implementation	R1	R2	R3	R4
Machado et al. [50]	An automatic policy Refinement Technique.	Translate SLA terms into a set of low-level rules. Finding the best path	Global, mostly Internet	Simulation	×	×	×	×
Bhattacharya and Das [51]	QoS enabled network architecture	Dynamic relation to bringing flexibility and adaptability and path optimization.	Global	Simulation	×	×	×	×
Körner et al. [52]	Conjoin WS-agreement standard and Open flow standard approach	WSAG4J framework	Global	Real Network implementation	✓	×	×	×

Moreover, there is no unique measurement technique available in SDN that can authenticate the provided services' satisfaction level. As a result, there is always a gap between the service provider and consumer to have service satisfaction information that can be considered for service provider selection decision-making and generate a lack of transparency among service providers and consumers. Besides, as shown in Table 7, none of the previous research proposes Service Oriented Architecture (SOA) based QoS delivery in SDN except [8,53]. As a result, no thought about personalized service selection and guaranteed QoS delivery of SDN in SOA based architecture has been given in the literature. Therefore, a very prominent research area of achieving QoS in SOA-based SDN has been left behind.

All the above comparative analysis (Tables 2–6) also demonstrates no unique quality of Service (QoS) measures standard available in SDN that can be adapted to SOA-based SDN architecture. While some researchers have focussed on this issue, the developed approaches need further work for SOA-based SDN to be applied in a production environment. SOA based SDN as a technology aims to provide clouds networking aspects such as "abstracted pool/grid of resources" and "elasticity" in the provision of SDN in a "service orientation" method [9]. After reviewing and discussing the above relevant existing research, the following noticeable shortcomings are identified which need to be addressed. We explain their importance in the use case scenario of a service user entering into an SLA agreement with a service provider.

1. There is no reputation data-driven SLA based approach that ensures QoS in SDNs. Having such a reputation-data driven SLA based approach can be used to select a suitable SDN service provider based on consumer requirements. This shortcoming needs to be addressed by developing an SLA based intelligent framework where the decision-making intelligence drives through some set of organization's reputations in delivering to the formed SLAs. This intelligent framework would leverage the service requester or consumer organization's opportunity in reputation data-driven suitable provider selection.
2. There is no existing method that enables service requester to personalize service delivery and to receive QoS as promised in SDN services. To ensure personalized service delivery, each service requester as an interacting party needs to determine its service requirements at a granular level based on which SLAs will be formed. The provision of a service provider delivering the service with personalized requirements crafts the service consumer's impression that the service provider considers them as a priority consumer and values their needs [23]. This notion has been considered in the literature that trust and personalized service delivery can be achieved and maintained only by consistently delivering the services according to the requirements that have been agreed upon. This can be achieved only if the service requirements determination and composition can be done by both parties comprehending in a granular manner before delivering the services.
3. There is no service satisfaction measurement technique that can verify the guarantee level committed to each service level objective defined in the service level agreement. The term "reliable" explicitly illustrates that there is no chance or a very limited chance of violating the defined Service Level Agreement (SLA) contract to a service provider and service consumer. As explained earlier, the violation of an SLA may involve significant financial penalties for both the parties. To avoid those scenarios, both service provider and consumer companies demand guaranteed,

Table 7
A comparative analysis among all existing QoS guaranteed approaches in SDN from the perspective of applying it in SOA-based SDN.

QoS management approaches	SDN Global Solution	SOA based SDN	R1	R2	R3	R4
Controller Design or Controller Placement approach	× (except [31–33])	× (except [8])	×	×	×	×
Resource Allocation or Dynamic Resource Allocation (Priority Users) approach	× (except [43]	×	×	×	×	× (except [40])
Queue Scheduling and Management approach	Mostly Global (except [41])	×	×	×	× (except [41])	× (except [41])
QoS Driven Routing approach	× (except [46–49])	×	×	×	× (except [45])	× (except [45–49])
SLA Based approach for delivering QoS	√	×	×	×	×	×

reliable service delivery. The existing studies discuss QoS guarantee in subjective based terms. That means that the QoS guarantee level has not been benchmarked and this cannot be used as a standardized measurement technique. Therefore, there is a need to have a unified approach that can explicitly measure the guarantee of the services quantitatively (objective-based). For this, we need to have a method that can compare the receiving service quality level against the service quality level that has been defined in SLA. To address this shortcoming, an intelligent approach is needed that evaluates the runtime services and therefore benchmarks the currently provided services with the committed services.

4. Very little research has been done that provides end-to-end QoS guarantee from an application being delivered in the SDN network. Existing research demonstrates that most SLA-based approaches are targeted to deliver the services or improve the solutions for the entire SDN network rather than specific application-based development. However, when SDN architecture is offered as a service, the focus should be on application based QoS. To achieve this, further research is required to overcome the drawbacks and other shortcomings illustrated here to make the best of it. Successful integration of SDN and NFV would be able to address the above research shortcomings partially. The possible solutions may include implementing the combination prototype of SDN and NFV architecture and testing, followed by recording the newly implemented combination network performance and benchmarking with further research for improvement.

According to the current business needs, the identified requirements R1–R4 need to be addressed to enable service provisioning. Without this, the realization of SOA-based SDN will remain in an infancy stage. In our future work, we aim to address these issues that will lead to providing QoS guaranteed SDN service delivery.

10. Conclusion

New computing technologies and platforms to deliver those technologies to meet humankind's evolving needs are always on the rise. To ensure that the delivered technologies meet the required or agreed quality, QoS management is an essential area of research in the literature. This paper focuses on the evolving service delivery platform of SOA-based SDNs and how to guarantee the promised QoS in this platform. In terms of QoS management, we first identified the difference between a traditional and SOA-based SDN which led us to define the four different requirements that QoS management techniques need to meet for them to be applied in SOA-based SDNs. We then categorize the relevant QoS management approaches in SDNs into five different categories before comparing the working of those approaches against the identified requirements. This has led them to define the open gaps for guaranteeing end-to-end QoS provisioning in SOA based SDN architecture. In our future work, we aim to develop algorithms to address these gaps to increase the adoption maturity of SOA-based SDNs.

CRediT authorship contribution statement

Shuraia Khan: Conceptualization of the idea, Formal analysis, Investigation, Data curation, Development, Initial draft version of the manuscript. **Farookh Khadeer Hussain:** Co-conceptualization of the idea, Led the project in setting its scope, Overall project and final draft review and feedback. **Omar K. Hussain:** Writing - review & editing, Feedback towards finalization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The third author acknowledges ARC LP160100080 for supporting his time on this work.

References

[1] Qiang Duan, Ninwan Ansari, Mehmet Toy, Software-defined network virtualization: an architectural framework for integrating SDN and NFV for service provisioning in future networks, IEEE Netw. 30 (5) (2016) 10–16.
[2] Murat Karakus, Arjan Durresi, Quality of service (QoS) in software defined networking (SDN): A survey, J. Netw. Comput. Appl. 80 (2017) 200–218.
[3] S. Khan, A. Gani, A.W.A. Wahab, M. Guizani, M.K. Khan, Topology discovery in software defined networks: Threats, taxonomy, and state-of-the-art, IEEE Commun. Surv. Tutor. 19 (1) (2017) 303–324, http://dx.doi.org/10.1109/COMST.2016.2597193.
[4] Mosab Hamdan, et al., A comprehensive survey of load balancing techniques in software-defined network, J. Netw. Comput. Appl. 174 (2021) 102856, http://dx.doi.org/10.1016/j.jnca.2020.102856.
[5] Muhammad Usman Younus, Saif ul Islam, Ihsan Ali, Suleman Khan, Muhammad Khurram Khan, A survey on software defined networking enabled smart buildings: Architecture, challenges and use cases, J. Netw. Comput. Appl. 137 (2019) 62–77, http://dx.doi.org/10.1016/j.jnca.2019.04.002.
[6] M. Priyanka, K.J. Singh, An idea for improvising the efficiency of SDN based business design with SOA, Int. J. Eng. Sci. Innovat. Technol. (IJESIT), Joirnal Articla 3 (3) (2014).
[7] Barbara Martini, Federica Paganelli, A service-oriented approach for dynamic chaining of virtual network functions over multi-provider software-defined networks, Fut. Internet 8 (2) (2016) 24.
[8] Qiang Duan, Network-as-a-service in software defined networks for end-to-end qos provisioning, in: The 23rd Wireless & Optical Communications Conference, IEEE, Newark, NJ, 2014, pp. 1–5.
[9] https://docplayer.net/4815829-White-paper-sdn-nfv.html.
[10] Slavica Tomovic, Neeli Prasad, Igor Radusinovic, SDN control framework for QoS provisioning, in: 2014 22nd Telecommunications Forum Telfor, TELFOR, IEEE, Belgrade, Serbia, 2014, pp. 111–114.
[11] Merilee Ford, Tim Stevenson, H.K.im Lew, Steve Spanier, Spanier Internetworking Technologies Handbook, Macmillan Publishing Co. Inc., 1997, p. 1128.
[12] I. Cisco Systems, Quality of Service Solutions Configuration Guide, in Congestion Avoidance Overview, Vol. 18, Cisco, 2014, (Online). Available: https://www.cisco.com/c/en/us/td/docs/ios/qos/configuration/guide/12_2sr/qos_12_2sr_book.html.

[13] https://www.cloudindustryforum.org/content/8-criteria-ensure-you-select-right-cloud-service-provider.
[14] Mohammed Alhamad, Tharam Dillon, Elizabeth Chang, Conceptual SLA framework for cloud computing, in: 2010 4th IEEE International Conference on Digital Ecosystems and Technologies, DEST, Dubai, United Arab Emirates, IEEE, 2010, pp. 606–610.
[15] Vincent C. Emeakaroha, Ivona. Brandic, Michael Maurer, Schahram Dustdar, Low level metrics to high level SLAs-LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments, in: 2010 International Conference on High Performance Computing and Simulation, HPCS, Caen, France, IEEE, 2010, pp. 48–54.
[16] Adil M. Hammadi, Omar Hussain, A framework for SLA assurance in cloud computing, in: 2012 26th International Conference on Advanced Information Networking and Applications Workshops, WAINA, Fukuoka-shi, Japan, IEEE, 2012, pp. 393–398.
[17] Yasmine M. Afify, Ibrahim F. Moawad, Nagwa L. Badr, Mohamed Tolba, Cloud services discovery and selection: Survey and new semantic-based system, in: A.E. Hassanien, T.-H. Kim, J. Kacprzyk, A.I. Awad (Eds.), Bio-Inspiring Cyber Security and Cloud Services: Trends and Innovations, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 449–477.
[18] Zia-ur Rehman, Omar Khadeer Hussain, Farookh Khadeer Hussain, User-side cloud service management: State-of-the-art and future directions, J. Netw. Comput. Appl. 55 (2015) 108–122, http://dx.doi.org/10.1016/j.jnca.2015.05.007.
[19] Walayat Hussain, Farookh Khadeer Hussain, Omar Hussain, Ravindra Bagia, Elizabeth Chang, Risk-based framework for SLA violation abatement from the cloud service provider's perspective, Comput. J. 61 (9) (2018) 1306–1322, http://dx.doi.org/10.1093/comjnl/bxx118.
[20] Zia ur Rehman, Omar Khadeer Hussain, Elizabeth Chang, Tharam Dillon, Decision-making framework for user-based inter-cloud service migration, Electron. Commer. Res. Appl. 14 (6) (2015) 523–531, http://dx.doi.org/10.1016/j.elerap.2015.08.002.
[21] Saurabh Kumar Garg, Steve Versteeg, Rajkumar Buyya, A framework for ranking of cloud computing services, Future Gener. Comput. Syst. 29 (4) (2013) 1012–1023, http://dx.doi.org/10.1016/j.future.2012.06.006.
[22] Walayat Hussain, Farookh Khadeer Hussain, Omar K. Hussain, Ernesto Damiani, Elizabeth Chang, Formulating and managing viable SLAs in cloud computing from a small to medium service provider's viewpoint: A state-of-the-art review, Inf. Syst. 71 (2017) 240–259, http://dx.doi.org/10.1016/j.is.2017.08.007.
[23] S. Khan, F.K. Hussain, A SOA based SLA negotiation and formulation architecture for personalized service delivery in SDN, in: International Conference on Network-Based Information Systems, Springer, 2019, pp. 108–119.
[24] http://thecustomerinstitute.blogspot.com/2013/05/satisfaction-has-legal-meaning.html ed.
[25] Anand V.A.kella, Kaiqi Xiong, Quality of service (QoS)-guaranteed network resource allocation via software defined networking (SDN), in: 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, DASC, Dalian, China, IEEE, 2014, pp. 7–13.
[26] Falak Nawaz, Mehdi Rajabi Asadabadi, Naeem Khalid Janjua, Omar Khadeer Hussain, Elizabeth Chang, Morteza Saberi, An MCDM method for cloud service selection using a Markov chain and the best-worst method, Knowl.-Based Syst. 159 (2018) 120–131, http://dx.doi.org/10.1016/j.knosys.2018.06.010.
[27] A.E. Youssef, An integrated MCDM approach for cloud service selection based on TOPSIS and BWM, IEEE Access 8 (2020) 71851–71865, http://dx.doi.org/10.1109/ACCESS.2020.2987111.
[28] Z. u. Rehman, O.K. Hussain, F.K. Hussain, Iaas cloud selection using MCDM methods, in: 2012 IEEE Ninth International Conference on E-Business Engineering, Hangzhou, China, 9-11 Sept. 2012, 2012, pp. 246–251, http://dx.doi.org/10.1109/ICEBE.2012.47.
[29] Kannan Govindarajan, Kong Chee Meng, Hong Ong, Wong Ming Tat, Sridhar Sivanand, Low Swee Leong, Realizing the quality of service (QoS) in software-defined networking (SDN) based cloud infrastructure, in: 2014 2nd International Conference on Information and Communication Technology, ICoICT, Bandung, Indonesia, IEEE, 2014, pp. 505–510.
[30] Bhakti Jadhav, Zia Saquib, Sanjay Pawar, Issues and parameters for improving QoS and performance in SDN, Int. J. Adv. Electron. Comput. Sci. 4 (7) (2017) (Online). Available: http://iraj.in.
[31] Tracy Yingying Cheng, Mengqing Wang, Xiaohua Jia, QoS-guaranteed controller placement in SDN, in: 2015 IEEE Global Communications Conference, GLOBECOM, San Diego, CA, IEEE, 2015, pp. 1–6.
[32] Hilmi E. Egilmez, S.Tahsin Dane, K.Tolga Bagci, A.Murat Tekalp, OpenQoS: An openflow controller design for multimedia delivery with end-to-end quality of service over software-defined networks, in: 2012 Asia-Pacific Signal & Information Processing Association Annual Summit and Conference, APSIPA ASC, Hollywood, California, IEEE, 2012, pp. 1–8.
[33] Davide Adami, Lisa Donatini, Stefano Giordano, Michele Pagano, A network control application enabling software-defined quality of service, in: 2015 IEEE International Conference on Communications, ICC, London, UK, IEEE, 2015, pp. 6074–6079.
[34] Chao Hu, Qun Wang, Xiuyue Dai, SDN over IP: enabling internet to provide better qos guarantee, in: 2015 Ninth International Conference on Frontier of Computer Science and Technology, FCST, Dalian, China, IEEE, 2015, pp. 46–51.
[35] Sachin Sharma, et al., Implementing quality of service for the software defined networking enabled future internet, in: 2014 Third European Workshop on Software Defined Networks, EWSDN, Budapest, Hungary, IEEE, 2014, pp. 49–54.
[36] Airton Ishimori, Fernando Farias, Eduardo Cerqueira, Antônio Abelém, Control of multiple packet schedulers for improving QoS on OpenFlow/SDN networking, in: 2013 Second European Workshop on Software Defined Networks, EWSDN, Berlin, Germany, IEEE, 2013, pp. 81–86.
[37] Kwangtae Jeong, Jinwook Kim, Young-Tak Kim, Qos-aware network operating system for software defined networking with generalized openflows, in: 2012 IEEE Network Operations and Management Symposium, NOMS, Maui, HI, IEEE, 2012, pp. 1167–1174.
[38] Andreas Kassler, Lea Skorin-Kapov, Ognjen Dobrijevic, Maja Matijasevic, Peter Dely, Towards QoE-driven multimedia service negotiation and path optimization with software defined networking, in: 2012 20th International Conference OnSoftware, Telecommunications and Computer Networks, SoftCOM, Split, Croatia, IEEE, 2012, pp. 1–5.
[39] Alba Xifra Porxas, Shih-Chun Lin, Min Luo, Qos-aware virtualization-enabled routing in software-defined networks, in: 2015 IEEE International Conference on Communications, ICC, London, IEEE, 2015, pp. 5771–5776.
[40] Melih A. Karaman, Burak Gorkemli, Sinan Tatlicioglu, Mustafa Komurcuoglu, Ozgur Karakaya, Quality of service control and resource prioritization with software defined networking, in: 2015 1st IEEE Conference on Network Softwarization, NetSoft, London, IEEE, 2015, pp. 1–6.
[41] Fuliang Li, Jiannong Cao, Xingwei Wang, Yinchu Sun, A QoS guaranteed technique for cloud applications based on software defined networking, IEEE ACCESS 5 (2017) 21229–21241.
[42] Cosmin Caba, José Soler, APIs for QoS configuration in software defined networks, in: 2015 1st IEEE Conference on Network Softwarization, NetSoft, London, U.K., IEEE, 2015, pp. 1–5.
[43] Sachin Sharma, et al., Demonstrating resilient quality of service in software defined networking, in: 2014 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS, Toronto, IEEE, 2014, pp. 133–134.
[44] Raphael Durner, Andreas Blenk, Wolfgang Kellerer, Performance study of dynamic QoS management for OpenFlow-enabled SDN switches, in: 2015 IEEE 23rd International Symposium on Quality of Service, IWQoS, Portland, OR, IEEE, 2015, pp. 177–182.
[45] Jochen W. Guck, Martin Reisslein, Wolfgang Kellerer, Model-based control plane for fast routing in industrial QoS network, in: IEEE 23rd International Symposium on Quality of Service, IWQoS, Portland, OR, IEEE, 2015, pp. 65–66.
[46] Yu-Jia Chen, Li-Chun Wang, Feng-Yi Lin, Bao-ShuhPaul Lin, Deterministic quality of service guarantee for dynamic service chaining in software defined networking, IEEE Trans. Netw. Serv. Manag. 14 (4) (2017) 991–1002.
[47] Faisal Alharbi, Zongming Fei, Improving the quality of service for critical flows in smart grid using software-defined networking, in: IEEE International Conference on Smart Grid Communications, SmartGridComm Sydney, Australia, IEEE, 2016, pp. 237–242.
[48] Peng Hongyu, Wang Weidong, Wang Chaowei, Chen Gang, Zhang Yinghai, QoS-guaranteed energy saving routing strategy using SDN central control for backbone networks, The Journal of China Universities of Posts and Telecommunications 22 (5) (2015) 92–100.
[49] Marian Seliuchenko, Orest Lavriv, Oleksiy Panchenko, Volodymyr Pashkevych, Enhanced multi-commodity flow model for QoS-aware routing in SDN, in: 2016 International Conference Radio Electronics & Info Communications, UkrMiCo, Kiev, IEEE, 2016, pp. 1–3.
[50] Cristian Cleder Machado, Lisandro Zambenedetti Granville, Alberto Schaeffer-Filho, Juliano Araujo Wickboldt, Towards SLA policy refinement for QoS management in software-defined networking, in: IEEE

28th International Conference on Advanced Information Networking and Applications, AINA Gwangju, Korea, IEEE, 2014, pp. 397–404.
[51] Bivas Bhattacharya, Debabrata Das, SDN based architecture for QoS enabled services across networks with dynamic service level agreement, in: 2013 IEEE International Conference on Advanced Networks and Telecommuncations Systems, ANTS, Kattankulathur, India, IEEE, 2013, pp. 1–6.
[52] Marc Körner, Alexander Stanik, Odej Kao, Applying QoS in software defined networks by using WS-agreement, in: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, CloudCom, Singapore, IEEE, 2014, pp. 893–898.
[53] Iris Bueno, Jose Ignacio Aznar, Eduard Escalina, Jordi Ferrer, J.A. Garcia-Espin, An OpenNaaS based SDN Framework for Dynamic QoS control, in: IEEE SDN for Future Networks and Services, SDN4FNS, Barcelona, Spain, 2013, pp. 1–7.

Shuraia Khan is a Ph.D. Candidate at the School of Computer Science, University of Technology Sydney. Australia. Her key research interests are in Software Defined Networking, Wireless Sensor Networks, trust-based computing, cloud computing, machine learning and data analytics. She has published widely in these areas in several conferences such as AINA, NBIS, ITNAC, IESC etc.

Dr. Farookh Khadeer Hussain is an Associate Professor in School of Software, University of Technology Sydney. He is an Associate Member of the Advanced Analytics Institute and a Core Member of the Centre for Artificial Intelligence. His key and research interests are in trust-based computing, cloud of things, blockchains machine learning. He has published widely in these areas in top journals such as FGCS, The Computer Journal, JCSS, IEEE Transactions on Industrial Informatics, IEEE Transactions on Industrial Electronics etc.

Omar Hussain is an Associate Professor at the University of New South Wales, Canberra. His research interests are in business intelligence, cloud computing and logistics informatics. In these areas, his research work focuses on utilizing decision making techniques for facilitating smart achievement of business outcomes. His research work has been published in various top international journals such as Information Systems, The Computer Journal, Knowledge Based Systems, Future Generation of Computer Systems etc. He has won awards and funding From competitive bodies such as the Australian Research Council for his research.

Evaluation of SLA Negotiation for Personalized SDN Service Delivery

Chapter · March 2020

DOI: 10.1007/978-3-030-44038-1_10

Evaluation of SLA Negotiation for Personalized SDN Service Delivery

Shuraia Khan[1], Farookh Khadeer Hussain[2]
[1] School of Computer Science, University of Technology Sydney
Shuraia.khan@student.uts.edu.au
[2] School of Computer Science, University of Technology Sydney
Farookh.hussain@uts.edu.au

Abstract. Ensuring Quality of Services (QoS) is crucial in service oriented business model. Service Level Agreement (SLA) is an essential agreement between a consumer and a provider which is a key element to sustain QoS. Service Negotiation is an initiation stage of SLA where the service requirements are reached with an agreements to avoid if there are any conflicting requirements. Guaranteeing QoS is one of the key challenge in SDN. Several intelligent solutions are proposed however most of them are application focused and unable to provide a personalized and reliable QoS delivery in SDN. This paper represent a reputation data driven SLA Negotiation framework that would be able to provide personalized and reliable service delivery in SDN and assist in QoS management for informed decision making. In addition, Fuzzy Interface Systems (FIS) is used to implement the framework and the results are discussed comprehensively in this paper.

Keywords: SLA Negotiation, QoS Negotiation, SDN

1 Introduction

Open Network Foundation (ONF) is defined Software Defined Network (SDN) as "*In SDN architecture, the control and data plane are decouple, network intelligence and state are logically centralized and the underlying network infrastructure is abstracted from the applications*."[1]. It is unquestionable that, SDN architecture breaks the vertical framework of the OSI mode meanwhile it keeps the promises of simplifying network operations and maintenance in efficient manner that improve on maximum utilization of network resources[2].

Service Oriented Architecture (SOA) offers an effective mechanism that offer flexible interaction among autonomous systems to meet diverse service requirements. As the principle of SOA is to provide a dynamic composition of services, several researchers approaching some evolutionary architecture in internetworking that targeted to achieve the SOA principle by composing several smaller services into specialized services[3]. Applications of the Service Orientation principle in SDN is propose to address the challenging problems of end to end QoS provisioning; such as this

principle leads the networking technology to the Network-as-a-service(NaaS) paradigm, that makes the future carrier networks look more like Clouds[4]

Service Level Agreement (SLAs) is essentials to the service-oriented Business Model whereas Service level delivery is an integral component of service level management. A good service delivery represent a good interaction between provider and client resulting with an increase on value[5]. On the other hand, business are managing service level agreements (SLA) to ensure that the defined services meets the certain level of criteria that is agreed upon agreement as well as ensuring the Quality of service (QoS) in terms of obligations, service pricing, penalties in case of agreement violations[6]. Service Negotiation is an initiation stage of SLA where the service requirements are reached with an agreements to avoid if there are any conflicting requirements.

Service Level agreement (SLA) is widely held process to manage product or service quality. We are working on to develop a SLA lifecycle in SDN for informed decision making and service negotiation is the first module of SLA lifecycle. In this research, we propose a SLA Negotiation Framework with the impression of SOA principals that offers personalized service delivery in SDN with demonstration of experimental results[7]. SLA negotiation framework is a unique framework that combines of five steps or process. The framework able to assist in intelligent decision making in terms of consumer selection as well as provider selection while selecting the required services. In addition, the framework is grounded with SOA context and able to fulfill various dimensions of service requirements that opening scope of personalized service delivery in SDN. Moreover, it uses data driven selection mechanism and decision making analogy to determine a successful service negotiation and develop SLA. The implementation details and experiment results are main focus in this paper.

This paper is organized as follows. Section 2 summarizes the existing studies. Section 3 explains the proposed SLA negotiation framework in SDN with SOA principle followed by validation and verification of the proposed framework in detail in section 4. The paper is concluded with future recommendation in section 5

2 Related Research

Service Level Agreement (SLA) driven Service negotiation frameworks to achieve Quality of Services (QoS) is an ongoing research. In that context, several service negotiation framework is developed in various service oriented domains, however in this paper, SLA driven Service negotiation framework in SDN are only discussed. A agent-mediation based quantitative SLA negotiation framework for internet is proposed that reduce the chance of rejection probability[8]. In this approach, based on three pre-defined user satisfaction criteria, the negotiation take place when network performances of the negotiation schemes are degraded.

A protocol similarity model and a similarity of virtual networks negotiation model is proposed where the best suited protocol identification performed to fulfill client's requirements[9]. A network resource allocation oriented SLA re-negotiation approach[10] and dynamic flow negotiation approach[11] in SDN is proposed to prevent resource idleness and meeting QoS requirements .
A real-life implementation oriented conjoint technique (WS-Agreement standard and Open Flow standard) technique in SDN is proposed by Korner et al.[12] This approach allows to define QoS requirements of the network and able to negotiate service level objectives. On the other hand, A SDN based self-customizable architecture is proposed [8] where the architecture offer personalizing network experience in cloud based SDN. However, this framework is not able to achieve the true benefit of personalizing service delivery. The concept personalization in this context is quite different according to the definition. The above literature survey represent that, there is very little research on SLA driven service negotiation in SDN. Moreover, there is no research on data driven service negotiation decision making. As a result, no experiment or validation is performed that help to determine how service negotiation strategies can take place to achieve QoS in SDN and able to assist in informed decision making. Moreover, personalize service delivery concept still remain unknown in SDN industry.

3. Proposed Service Oriented Service Negotiation Framework

"Service Oriented based SLA negotiation" is an intelligent framework where the criteria of the required services is predefined, agreed in presents with third party before SLA formulation decision made. Thus, the service consumer have a broader scope to receive personalize service delivery using this intelligent framework.
The main two key features that the framework offering are, assist in intelligent decision making of selecting suitable service provider considering their previous transaction history. Another key feature is, assist in intelligent decision making for service provider whether the service request should accept or reject in terms of providers' circumstances. These two feature helps to build trust relationship among Consumer Company and the service provider company as they both have very detail knowledge about each other before the decision made. .
Service criteria specification, service provider selection, suitable consumer selection and Service Level Agreement formulation are four key tasks in the SLA framework that involves five stages as follows[7], detail discussion are available in our another paper[7]:
Stage1: Consumer send a service request (To the registry)
Stage2: Service request received by Intermediate agent (registry/broker).
Stage3: Interested Service providers reply with their interest (evaluate service request and intelligent decision will made).
Stage4: Requester select the suitable Service Provider (Accept or Rejection decision).
Stage5: Service Level Agreement Formulation upon brief service negotiation

3.1 Stage1: Consumer send a service request (To the registry)

At this stage, the requester send a service request with their service requirements. The request can be represent in a draft form that include service specification details such as: service items, availability, duration, quality specification etc. The requester sends the service request draft to the registry or broker or service agent. As an example; sending an open tender for a service[7].

3.2 Stage2: Service request received by Intermediate agent (registry/broker).

At this stage, the intermediate agent (service agent / registry / broker) receives service request and then send the service request to the service providers. We assume that, intermediate agent send the request (with service specification) to providers who are registered with that agent. On the other hand, the intermediate agent sends an acknowledgement with working progress[7] to the requester.

3.3 Stage3: Interested Service providers reply with their interest (evaluate service request and intelligent decision will make).

At this stage, service provider take some intelligent decision of finding their suitability to accept the service request considering several factors. To take the decision, the following factors are considered:
Reputation rating of the Consumer to measure reliability: Factor1
Duration of the Services they require: Factor2
Service Risk Propensity: Factor3

3.3.1 Reputation Rating of the consumer: Factor1

In this framework, historical reputation rating of the consumer data is considered to measure the reliability. The dataset screen shots are provided in the next section. For this framework, last few years reputation rating is considered. Two variations of reputation rating data is accepted such as; company's overall reputation rating and other provider's satisfaction rating about this company. The mathematical equation is providing below and the detail discussion of the equation are shown in our another paper[7]. Reputation rating scale is defined as1 to 5.

$$R=5\times ((R \text{ rating, OPR rating})/(\text{Max R rating, Max OPR rating})) \quad (1)$$

3.3.2 Duration of the Services: Factor2

The duration of the services that the requester (consumer) needs is the second parameter we have used to develop the framework. A rating formula is used to represent service request duration. This framework provide scope for the service

provider to change their duration rating scale according to company demand. The rating scale detail is discussed in our another paper[7].

3.3.3 Factor 3 or F3: Risk propensity of accepting a contract

Risk Propensity is another important factor that is considered while calculating the intelligent decision about service provider's suitability. Risk propensity refers to evaluate the company risk before accepting the service request. This approach assist the provider to evaluate their risk and have options to define their risk range based on company requirements and structure. Further detail of Risk propensity rating equation is discussed in our another paper[7] . In this research, Risk is defined as low risk, medium, risk and then high risk and the scale is defined as 1 to5.

$$\text{Risk R} = \int \left(\frac{Current\ Risk\ Level}{Maximum\ Risk} \right) \quad (2)$$

3.3.4 Suitability calculation:

At this stage, we build an reputation data driven intelligent system that able to determine the suitability of the provider to accept the service request. Fuzzy interface systems is used to map input variables for the output value. Mamdani is a very popular Fuzzy Interface Systems that is used to calculate the suitability of the provider and assist in very important intelligent decision making. The implementation results are demonstrated later in this paper.

3.4 Stage4: Requester select the suitable Service Provider (Accept or Rejection decision).

The outcomes of previous stage (Stage 3) generates a list of provider who are interested to provide the services. At this stage, the consumer also has opportunity to select the suitable service provider that they feel confident, in other word is "selecting suitable provider for their requested service". We assume that, providers are already in the market and running their business successfully from few years and have existing transaction records.

3.4.1 Suitable Provider Selections:

We assume that, the providers already have previous transaction history and have their own profile that are stored in a repository. The repository is collected and maintained by third party service agent. The service requester can access the repository and able to access the profiles of the interested providers. In this stage, the consumer need to select a suitable provider that they fell confident to go with. This framework provide a reputation based provider selection technique that assist the consumer to make an intelligent decision in terms of suitable provider selection. Our previous paper disused about this framework in detail[7] however we have improved this framework and implement in a Fuzzy Interface System and received the results. The experiment details is discussed in next section.

If a provider x, has performed Ts number of successful transaction among Tn number of total transaction, we can calculate the providers transaction trend using following equation[13].

$$Ts/Tn \times 100 = \text{Transaction trend.} \quad (3)$$

3.5 Stage5: Service Level Agreement Formulation upon brief service negotiation

At this stage, Service provider and consumer are agreed in a common agreement and make a formal commitment. At this point, the service provider already have a clear understanding about the service requester company from their company profile. The requester company also have comprehensive knowledge about Provider Company. In this stage, the requester make a formal document that includes detail definition of the services that they require, in other words defining QoS requirements. This document allow the requester to define their service requirements with fine granulated specifications and offers to indicate differentiated service request if required (services for different applications with various requirements) with their expectation standard of each services. After reviewing the service request draft, the provider obtain a comprehensive understanding about the requested services. By obtaining requester Company's background, their service requirements and expectations that are very clearly demarcated in the document, therefore, personalized service delivery feature can be achieved by provider. Furthermore, the service provider and requester company negotiate several other business objectives such as: service cost, SLA violation penalties etc. followed by agreeing from both parties and SLA is finalized and formulated that is signed by both parties in the presence of third parties.

4 Validation and Verification

To validate the proposed "Service Oriented based Service Negotiation Framework", we carry out a simulation representing two consumers that request services from a provider. Our intension is, in the simulation, we will represent that how this Service Negotiation framework assists the SDN service provider to first determine which consumers' service request should accept according to their company reputation and transaction trend value. We also will represent, our proposed framework assist in intelligent decision making for the SDN consumer in terms of selecting suitable Service provider. To validate this framework, we introduce two sets of new data. One of the dataset is SDN service provider reputation rating and other one is SDN Consumers' reputation rating. The source of these datasets and collection procedures the discussion below.

4.1 Data sets

For this research, we have mostly used Synthetic dataset. The reason of creating a synthetic dataset is, there no source from where we could collect consumer reputation

data. Synthetic data is very important and useful when either privacy needs limit the availability or usage of the data or when the data needed for a test environment simply does not exist[14]. Two sets of datasets is used to implement this framework. SDN service provider reputation rating is the first dataset that we have collected from Software Defined Networking trust radius website. This trust rating site mainly provide top 20 SDN vendor' or service providers' reputation review, overall rating, their available SDN products and reviewer's Company Size. This trust radius site also giving opportunity to view each SDN products reputation and each vendors' reputation separately. All of the reviewers are vetted reviewers and verified company users and the reviews are performed out of 10. We have collected customized information to make our raw SDN service provider reputation rating dataset and we scale them out of 5.

To implement our framework, we require additional features and data in this data set such as number of successful transaction, total number of transactions, establishment year etc. Other than reputation rating, rest of the information we have developed as such the data set can perform as a synthetic dataset. The snapshot of the data set is providing below.

Service Provider	establishment year	Reputation rating	Total number of transaction Tn	Total number of successful transaction s Ts	Transaction Trend(Ts/Tn)*100	Esulidean Distance provider 7	Rank
1	2013	4.8	17	16	94.1	16.155	16.0000
2	2014	4.7	15	10	66.7	13.928	14.0000
3	2014	3.3	8	5	62.5	7.810	8.0000
3	2016	3.1	9	6	66.7	7.616	7.0000
4	2014	4.2	17	12	70.6	15.811	15.0000
5	2017	2.8	19	10	52.6	17.117	17.0000
6	2013	4.1	25	16	64.0	23.770	20.0000
7	2019	0	2	0	0.0	0.000	1.0000
8	2019	2.2	8	5	62.5	6.000	5.0000
9	2018	3.7	14	9	64.3	12.042	11.0000
10	2016	4.6	22	19	86.4	20.224	19.0000
11	2015	1.5	21	5	23.8	19.416	18.0000
13	2016	2.3	7	5	71.4	5.831	4.0000
14	2017	3	13	9	69.2	11.180	10.0000
15	2019	2.6	6	3	50.0	4.000	2.0000
16	2018	4.3	7	5	71.4	5.099	3.0000
17	2018	3.9	9	5	55.6	7.071	6.0000
18	2017	1.8	11	4	36.4	9.220	9.0000
19	2016	3.7	14	11	78.6	12.369	12.0000
20	2017	3.2	15	11	73.3	13.153	13.0000

Fig 1. Service provider reputation profile dataset

The second dataset that we have developed is consumer reputation rating dataset and the data set is fully synthetic dataset. As the real data does not exist in our circumstances, synthetic data is the only solution. The Consumer rating dataset snapshot are providing below.

SDN consumer Rating rating dataset					5*(R/Max((OPR/MAXOPR)
SDN Consumer	Company Rating R	Max R	Other Provider rating OPR	Max OPR	Reliability 1
1	4.3	5	4.1	5	3.5
2	3.2	5	3.5	5	2.2
3	2.1	5	2.4	5	1.0
5	2.4	5	2.1	5	1.0
6	1.1	5	2.7	5	0.6
7	4.4	5	3.8	5	3.3
8	3.8	5	3	5	2.3
9	2.8	5	1.8	5	1.0
10	4.5	5	4.1	5	3.7
11	3.6	5	3.8	5	2.7
12	1	5	1.8	5	0.4
13	2.9	5	2.1	5	1.2
14	4.7	5	4.1	5	3.9
15	3.4	5	3.9	5	2.7
16	2.5	5	2.2	5	1.1
17	1.8	5	2.2	5	0.8
18	3.9	5	4.9	5	3.8
19	4.2	5	3.9	5	3.3
20	4.9	5	1.9	5	1.9

Fig 2. Service consumer reputation rating dataset

4.2 Experiment Results and Discussion

To evaluate the proposed service negotiation framework, we have used MATLAB Fuzzy Interface System. In the framework, we have seen that there are two major decision making functions in the framework, firstly the service provider will decide whether they will accept or reject the service request. The developed method assist in this decision making by considering consumer companies reputation rating. Whereas the second decision will made by consumer to select which provider is most reliable for them. The implementation details are discussing below.

4.2.1 Service provider decision making to accept or reject the request:

To evaluate this approach we have used three fuzzy input named "Service Duration", Service Risk and Consumer Company Reliability. We also setup the fuzzy output names "Suitability of the consumer". We define the membership functions of these three inputs. We have used Mamdani approach to define the rules using fuzzy association techniques. We generate 60 combination of rules. To validate the approach, we used our dataset that we have developed.

The figure 3 represents a 3 dimensional surface view where the x and y inputs are service risk and service duration. Output z represents suitability of the consumer. The figure 3 demonstrate that, considering only two criteria, if the service risk is high and service duration is short, then the decision of suitability of that requester consumer is very low. Besides, the figure 4. below represents another surface view where x is taken as consumer company reliability input, y is taken as service risk input and z is generate the suitability of the consumer output. The figure 4 clearly demonstrate that based on considering two criteria only, if the service request generate mid-level service risk for the provider company and if the service requester company reliability

is medium to high, then our proposed approach generate decision as highly suitable consumer.

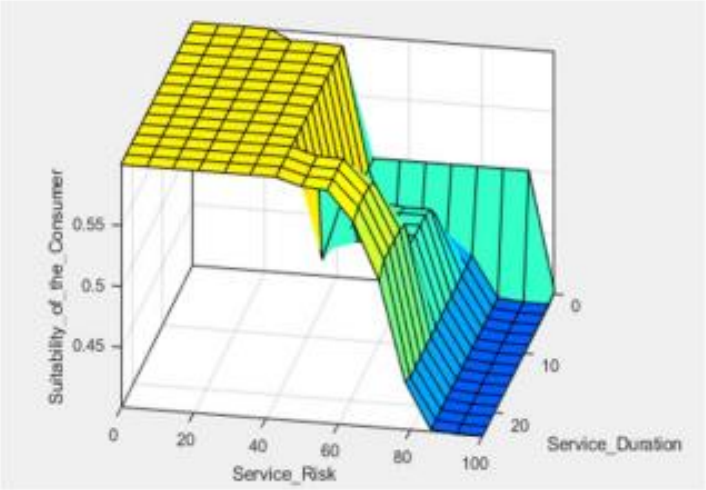


Fig 3. 3 dimensional surface view where the x input is service risk and y input is service duration and z output is suitability of the consumer.

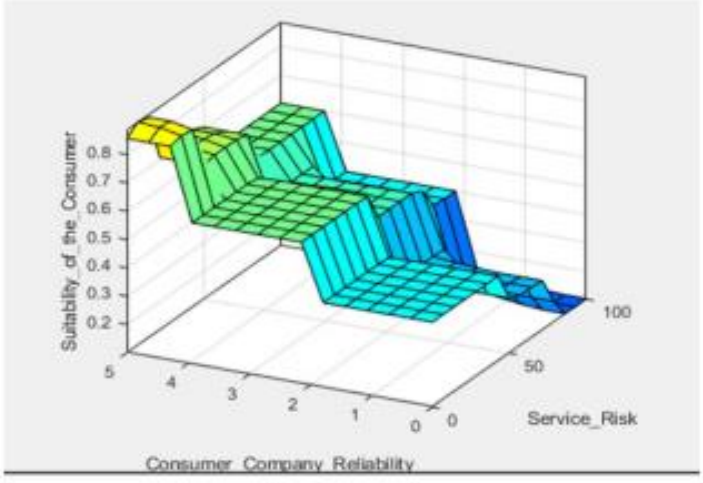


Fig 4. 3 dimensional surface view where the x input is Consumer Company Reliability and y input is service risk and z output is suitability of the consumer.

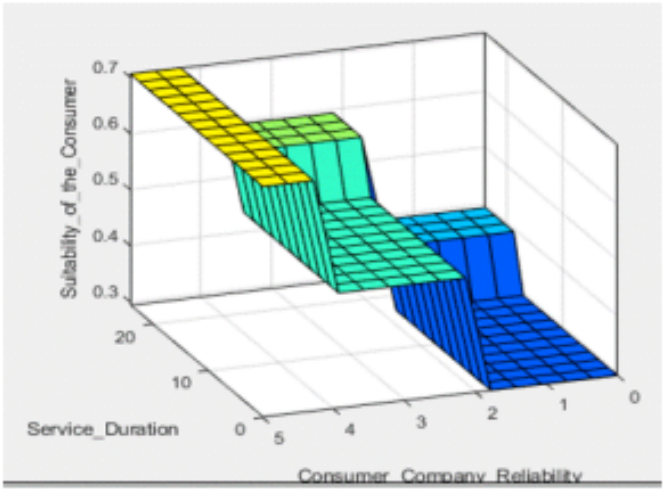


Fig 5. 3 dimensional surface view where the x input is Service Duration and y input is Consumer Company Reliability and z output is suitability of the consumer.

4.2.3: Service Consumer selecting their suitable provider

To evaluate the reputation based provider selection approach, we have used two fuzzy input named "Reputation rating "and "Transaction trend". We also setup the fuzzy output names "Accept/Reject Decision". We define the membership functions of these two inputs. As the decision will be the simple accept or reject the provider, we have defined only two membership functions for output. We have used Mamdani Fuzzy Interface Systems and define the rules using fuzzy association techniques. We generate 9 combination of rules this time. To validate the approach, we have used our service provider profile history dataset that we have developed.

The input "transaction trend" data is calculated using equation 3 above where we have considered total number of transaction and number of successful transaction data of each provider. The figure 6. Represents a 3 dimensional surface view of provider selection decision making where the x and y inputs are Transaction trend and reputation rating. Output z represents accept or reject decision. We have define the output membership function range as 0 to 47% get rejected and 40% - 100% get accepted. The figure 6. Demonstrate that, considering two criteria, if the transaction trend is high and reputation rating is medium, then the decision support the acceptance.

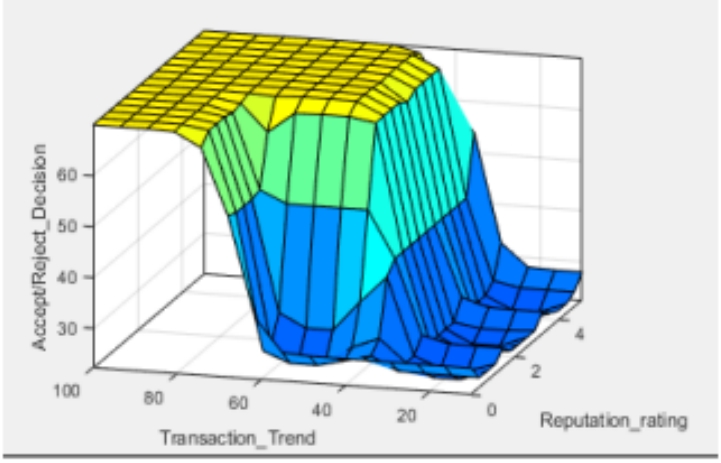


Fig 6. 3 dimensional surface view where the x input is Transaction Trend and y input is Reputation rating and z output is Accept/Reject decision.

5 Conclusion and Future Recommendation

SLA management is a vital challenge today for a service oriented business model today, particularly in SDN business environment. In this paper we introduced a service negotiation framework with experiment result which is an initial part of SLA. The findings of literature review demonstrate that there is no research in managing SLA to ensure QoS in SDN[7]. Moreover, there is no or very little research on SOA based service management in SDN that able to provide personalized service delivery. SLA Negotiation framework introduced to address these gaps as the service negotiation take place before the SLA developed to avoid any conflicting situation. In such, the framework able to achieve the personalized service delivery in SDN which is an advance feature of SOA. Moreover, this research offered a reputation rating based service provider selection as well as service consumer selection that is an ongoing challenge for service provider and consumer to select a trustworthy company.

This Service Negotiation framework in SDN introduce a pathway of building a communication between service consumer and service provider that assist in developing a trust relationship among them. This experiment result demonstrate the success of the framework using Artificial Intelligence (AI) systems. There are few subsections of SLA

References

1. Karakus, M. and A. Durresi, *Quality of service (qos) in software defined networking (sdn): A survey*. Journal of Network and Computer Applications, 2017. **80**: p. 200-218.
2. Yu, C., et al., *QoS-aware traffic classification architecture using machine learning and deep packet inspection in SDNs*. Procedia computer science, 2018. **131**: p. 1209-1216.
3. Martini, B. and F. Paganelli, *A service-oriented approach for dynamic chaining of virtual network functions over multi-provider software-defined networks*. Future Internet, 2016. **8**(2): p. 24.
4. Duan, Q., N. Ansari, and M. Toy, *Software-defined network virtualization: an architectural framework for integrating SDN and NFV for service provisioning in future networks*. IEEE Network, 2016. **30**(5): p. 10-16.
5. *What Is Service Delivery?* ; Available from: https://www.reference.com/business-finance/service-delivery-b40d5bbd6275c5da.
6. Emeakaroha, V.C., et al. *Low level metrics to high level SLAs-LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments*. in *High Performance Computing and Simulation (HPCS), 2010 International Conference on*. 2010. IEEE.
7. Khan, S. and F.K. Hussain. *A SOA Based SLA Negotiation and Formulation Architecture for Personalized Service Delivery in SDN*. in *International Conference on Network-Based Information Systems*. 2019. Springer.
8. Chieng, D., A. Marshall, and G. Parr, *SLA brokering and bandwidth reservation negotiation schemes for QoS-aware internet*. IEEE Transactions on Network and Service Management, 2005. **2**(1): p. 39-49.
9. Gomes, R.L., L.F. Bittencourt, and E.R. Madeira. *A similarity model for virtual networks negotiation*. in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. 2014. ACM.
10. Gomes, R.L., L.F. Bittencourt, and E.R. Madeira. *SLA Renegotiation According to Traffic Demand*. in *2nd Workshop on Network Virtualization and Intelligence for the Future Internet (WNetVirt)*. 2013.
11. Ghalwash, H. and C. Huang. *A QoS Framework for SDN-Based Networks*. in *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)*. 2018.
12. Körner, M., A. Stanik, and O. Kao. *Applying QoS in Software Defined Networks by Using WS-Agreement*. in *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*. 2014. IEEE.
13. Walayet Hussain, F.K.H., Omar Khadeer Hussain, EliZabeth Chang, *Provider-Based Optimized Personalized Viable SLA(OPV-SLA) Framework to Prevent SLA violation*. The Computer Journal Advance Access, 2016. **Section A**.
14. *Synthetic Data: An Introduction & 10 Tools [2019 Update]*. 2019; Available from: https://blog.aimultiple.com/synthetic-data/.