UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA

Departamento de Ingeniería del Software e Inteligencia Artificial

TESIS DOCTORAL

Modelo de conciencia situacional para el análisis de datos en redes móviles 5G: arquitectura SELFNET

A situational awareness model for data analysis on 5G mobile networks: the SELFNET analyzer framework

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Lorena Isabel Barona López

Directora

Luis Javier García Villalba

Madrid, 2018

Modelo de Conciencia Situacional para el Análisis de Datos en Redes Móviles 5G: Arquitectura SELFNET

A Situational Awareness Model for Data Analysis on 5G Mobile Networks: The SELFNET Analyzer Framework

Thesis by

Lorena Isabel Barona López

In Partial Fulfillment of the Requirements for the Degree of
Doctor por la Universidad Complutense de Madrid en el
Programa de Doctorado en Ingeniería Informática

Advisor

Luis Javier García Villalba

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Madrid, April 2017

A Situational Awareness Model for Data Analysis on 5G Mobile Networks: The SELFNET Analyzer Framework

Thesis by

Lorena Isabel Barona López

In Partial Fulfillment of the Requirements for the Degree of
Doctor por la Universidad Complutense de Madrid en el
Programa de Doctorado en Ingeniería Informática

Advisor

Luis Javier García Villalba

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Madrid, April 2017

Modelo de Conciencia Situacional para el Análisis de Datos en Redes Móviles 5G: Arquitectura SELFNET

TESIS DOCTORAL

Memoria presentada para obtener el título de
Doctor por la Universidad Complutense de Madrid
en el Programa de Doctorado en Ingeniería Informática

Lorena Isabel Barona López

Director

Luis Javier García Villalba

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Madrid, Abril de 2017

Dissertation submitted by Lorena Isabel Barona López to the *Departamento de Ingeniería del Software e Inteligencia Artificial* of the *Universidad Complutense de Madrid* in Partial Fulfillment of the Requirements for the Degree of *Doctor por la Universidad Complutense de Madrid en el Programa de Doctorado en Ingeniería Informática.*

Madrid, 2017.

(Submitted April 24, 2017)

Title:

A Situational Awareness Model for Data Analysis on 5G Mobile Networks: The SELFNET Analyzer Framework

PhD Student:

Lorena Isabel Barona López (lorebaro@ucm.es)
Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid
28040 Madrid, Spain

Advisor:

Luis Javier García Villalba (javiergv@fdi.ucm.es)

This work has been done within the Group of Analysis, Security and Systems (GASS, http://gass.ucm.es/), Research Group 910623 from the Universidad Complutense de Madrid (UCM) as part of the activities of the research project funded by the European Commission Horizon 2020 Programme under Grant Agreement number H2020-ICT-2014-2/671672-SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks). This research has also been supported by Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación SENESCYT - Convocatoria Abierta 2013 (Quito, Ecuador). Part of this work was done during my stay in Portugal at University of Aveiro (Institute of Telecommunications).

Tesis Doctoral presentada por la doctoranda Lorena Isabel Barona López en el Departamento de Ingeniería del Software e Inteligencia Artificial de la Universidad Complutense de Madrid para la obtención del título de Doctor por la Universidad Complutense de Madrid en el Programa de Doctorado en Ingeniería Informática.

Terminada en Madrid el 24 de Abril de 2017.

Título:

Modelo de Conciencia Situacional para el Análisis de Datos en Redes Móviles 5G: Arquitectura SELFNET

Doctorando:

Lorena Isabel Barona López (lorebaro@ucm.es)
Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid
28040 Madrid, España

Director:

Luis Javier García Villalba (javiergv@fdi.ucm.es)

Esta tesis doctoral ha sido realizada dentro del grupo de investigación GASS (Grupo de Análisis, Seguridad y Sistemas, grupo 910623 del catálogo de grupos reconocidos por la UCM) como parte de las actividades del proyecto de investigación SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks) financiado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 (H2020-ICT-2014-2/671672-SELFNET). Asimismo, el presente trabajo ha sido financiado por la Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación SENESCYT - Convocatoria Abierta 2012 (Quito, Ecuador). Parte de esta investigación ha sido realizada en el Instituto de Telecomunicaciones de la Universidad de Aveiro.

This thesis is dedicated to my family:
They are my light and my strength

Dedico esta tesis a mi familia:
Ellos son mi luz y mi fuerza

Acknowledgments

First, I would like to thank my supervisor, Luis Javier García Villalba for the continuous support and patience during my research. I have learned a lot of with his guidance.

Thanks to all the members of the GASS research group of the Universidad Complutense de Madrid.

I would like to thank to my friends, especially Marco, Leonardo and Jorge. Thanks for their support and stimulating discussions during this time.

Finally, I would like to express my thanks to my family, Martha, Alfredo, Gissela, Pablo, Efraín, Eulalia, Paulina, Carolina, Gustavo, Ximena and Patricio, for their support and confidence. They always encouraged me during my years of study and have taught me to work hard to achieve my goals. This accomplishment would not have been possible without them. Thank you.

This work has been done within the Group of Analysis, Security and Systems (GASS, http://gass.ucm.es/), Research Group 910623 from the Universidad Complutense de Madrid (UCM) as part of the activities of the research project funded by the European Commission Horizon 2020 Programme under Grant Agreement number H2020-ICT-2014-2/671672-SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks). This research has also been supported by Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación SENESCYT - Convocatoria Abierta 2013 (Quito, Ecuador). Part of this work was done during my stay in Portugal at University of Aveiro (Institute of Telecommunications).

Agradecimientos

En primer lugar quiero agradecer a mi director, Luis Javier García Villalba por su apoyo continuo y por su paciencia durante el desarrollo de mi investigación. He aprendido mucho con su guía.

Muchas gracias a los miembros del grupo de investigación GASS de la Universidad Complutense de Madrid por todas las facilidades ofrecidas.

Quiero dar las gracias a mis amigos, especialmente a Marco, Leonardo y Jorge. Muchas gracias por su apoyo y por las interesantes discusiones durante este tiempo.

Finalmente, me gustaría expresar mi agradecimiento a mi familia, Martha, Alfredo, Gissela, Pablo, Efraín, Eulalia, Paulina, Carolina, Gustavo, Ximena y Patricio, por su apoyo y confianza. Mi familia siempre me ha alentando durante mis años de estudio y me ha enseñado a trabajar duro para alcanzar mis metas. Este logro no habría sido posible sin ellos. Mil gracias por ser parte de mi vida.

Esta tesis doctoral ha sido realizada dentro del grupo de investigación GASS (Grupo de Análisis, Seguridad y Sistemas, grupo 910623 del catálogo de grupos reconocidos por la UCM) como parte de las actividades del proyecto de investigación SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks) financiado por la Comisión Europea dentro del Programa Marco de Investigación e Innovación Horizonte 2020 (H2020-ICT-2014-2/671672-SELFNET). Asimismo, el presente trabajo ha sido financiado por la Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación SENESCYT - Convocatoria Abierta 2012 (Quito, Ecuador).

Contents

List of Figures

List of Tables

List of Acronyms

3GPP	*Third Generation Partnership Project*
5G	*Fifth Generation*
5GMF	*Fifth Generation Mobile Communications Promotion Forum*
AA	*Active Application*
ABNO	*Application Based Network Operation*
ADB	*Aggregated Data Bundle*
AMQP	*Advanced Message Queuing Protocol*
ANN	*Artificial Neural Network*
API	*Application Programming Interface*
ARPU	*Average Revenue Per User*
BGP	*Border Gateway Protocol*
BN	*Bayesian Network*
BSS	*Business Support System*
BWAC	*Broadband Wireless Access and Applications Center*
BWRC	*Berkeley Wireless Research Center*
CLI	*Command Line Interface*
CPE	*Customer Premise Equipment*
D2D	*Device to Device Communication*

DDoS	*Distributed Denial of Service*
DNS	*Domain Name System*
DPI	*Data Packet Inspection*
DWDM	*Dense Wavelength Division Multiplexing*
EE	*Execution Environment*
EMS	*Element Management System*
EPC	*Evolved Packet Core*
ETSI	*European Telecommunications Standards Institute*
FDL	*Floodlight*
FIFO	*First In First Out*
ForCES	*Forwarding and Control Element Separation*
FRP	*Functional Reactive Programming*
HoN	*Health of Network*
IaaS	*Infrastructure as a Service*
ICN	*Information Centric Networking*
IDS	*Intrusion Detection System*
IETF	*Internet Engineering Task Force*
IMS	*IP Multimedia Subsystem*
IoT	*Internet of Thing*
IPR	*Intellectual Property Right*
ISRA	*Information Security Risk Assessment*
ISRM	*Information Security Risk Management*
ITS	*Issue Tracking System*

KPI	*Key Performance Indicator*
LTE	*Long Term Evolution*
M2M	*Machine to Machine Communication*
MCN	*Mobile Cloud Network*
MDSE	*Model driven Software Engineering*
MEC	*Mobile Edge Computing*
MFC	*MobileFlow Controller*
MFFE	*MobileFlow Forwarding Engine*
MIMO	*Multiple Input Multiple Output*
MLN	*Markov Logic Network*
MME	*Mobile Management Entity*
MOS	*Mean Opinion Score*
NaaS	*Network as a Service*
NAT	*Network Address Translation*
NETCONF	*Network Configuration Protocol*
NF	*Network Function*
NFaaS	*Network Functions-as-a-Service*
NFV	*Network Function Virtualization*
NFVI	*Network Function Virtualization Infrastructure*
NFVIaaS	*NFV Infrastructure as a Service*
NFVO	*Network Function Virtualization Orchestrator*
NIC	*Network Interface Card*
NMS	*Network Management System*
NodeOS	*Node Operating System*
NOS	*Network Operating System*

NSSA	*Network Security Situational Awareness*
ODL	*OpenDaylight*
ONF	*Open Networking Foundation*
OS	*Operating System*
OSPF	*Open Shortest Path First*
OSS	*Operational Support System*
OTT	*Over the Top*
PaaS	*Platform as a Service*
PAF	*Path Assignment Function*
PGW	*Packet Data Network Gateway*
QFF	*QoE Fairness Framework*
QMOF	*QoS Matching and Optimization Function*
QoE	*Quality of Experience*
QoS	*Quality of Service*
RAN	*Radio Access Network*
RAT	*Radio Access Technology*
RCP	*Routing Control Platform*
RFC	*Request for Comment*
SA	*Situational Awareness*
SBC	*Session Border Controller*
SC	*Self-Congestion*
SDMN	*Software-Defined Mobile Network*
SDN	*Software Defined Networking*

SDO	*Standards Developing Organization*
SDR	*Software Defined Radio*
SELFNET	*Self-Organized Network Management in Virtualized and Software Defined Networks*
SF	*Service Function*
SFC	*Service Function Chaining*
SG	*Self-Guard*
SGW	*Serving Gateway*
SLA	*Service Level Agreement*
SME	*Small Medium Enterprise*
SON	*Self Organizing Networks*
SSA	*Shared Situational Awareness*
SVM	*Support Vector Machine*
TCO	*Total Cost of Ownership*
TTM	*Time to Market*
VIM	*Virtual Infrastructure Manager*
VLAN	*Virtual Local Area Network*
VM	*Virtual Machine*
VN	*Virtual Network*
VNF	*Virtual Network Function*
VNF-FG	*VNF Forwarding Graph*
VPN	*Virtual Private Network*

Abstract

5G networks hope to provide a secure, reliable and high-performance environment with minimal disruptions in the provisioning of advanced network services, regardless the device location or when the service is required. This new network generation will be able to deliver ultra-high capacity, low latency and better Quality of Service (QoS) compared with current Long Term Evolution (LTE) networks. In order to provide these capabilities, 5G proposes the combination of advanced technologies such as Software Defined Networking (SDN), Network Function Virtualization (NFV), Self-organized Networks (SON) or Artificial Intelligence. In particular, 5G will be able to face unexpected changes or network problems through the identification of specific situations, taking into account the user needs and the Service Level Agreements (SLAs).

Nowadays, the main telecommunication operators and community research are working in strategies to facilitate the data analysis and decision-making process when unexpected events compromise the health in 5G Networks. Meanwhile, the concept of Situational Awareness (SA) and incident management models applied to 5G Networks are also in an early stage. The key idea behind these concepts is to mitigate or prevent harmful situations in a reactive and proactive way. In this context, Self-Organized Network Management in Virtualized and Software Defined Networks Project (SELFNET) combines SDN, NFV and SON concepts to provide a smart autonomic management framework for 5G networks. SELFNET resolves common network problems, while improving the QoS and Quality of Experience (QoE) of end users.

For this purpose, in this thesis the applicability of the three stages of processing information of Endsley Situational Model is presented. This approach takes into account traditional management guidelines and the dynamicity of 5G environments in order to lay the foundation of a Situational Awareness (SA) system. SELFNET situational awareness model is able to know what is happening with the different network elements and how to optimize the network performance, fix or prevent failures.

A key aspect of the proposed approach is the contextual analysis of the information gathered from the network devices, which is an emergent topic in 5G networks. As a result, this thesis also proposes the SELFNET Analyzer Framework to diagnosis the network state and predict possible problems in order to facilitate the decision-making process in 5G Networks. This proposal provides pattern recognition, reasoning and prediction capabilities to infer suspicious behaviours and then facilitate reactive and proactive responses. Furthermore, SELFNET Analyzer Framework follows a use case driven approach, where an operator is able to customize the parameters, the analysis functions and the rules taken into account in the diagnosis process. The result of Analyzer framework is a scalable and simple solution able to meet 5G requirements.

Keywords: 5G Incident Management, Data Analysis, Network Function Virtualization, SELFNET, Software Defined Networking, Self-Organizing Networks, Situational Awareness.

Resumen

Se espera que las redes 5G provean un entorno seguro, confiable y de alto rendimiento con interrupciones mínimas en la provisión de servicios avanzados de red, sin importar la localización del dispositivo o cuando el servicio es requerido. Esta nueva generación de red será capaz de proporcionar altas velocidades, baja latencia y mejor Calidad de Servicio (QoS) comparado con las redes actuales Long Term Evolution (LTE). Para proveer estas capacidades, 5G propone la combinación de tecnologías avanzadas tales como Redes Definidas por Software (SDN), Virtualización de las Funciones de Red (NFV), Redes auto-organizadas (SON) e Inteligencia Artificial. De manera especial, 5G será capaz de solucionar o mitigar cambios inesperados o problemas típicos de red a través de la identificación de situaciones específicas, tomando en cuenta las necesidades del usuario y los Acuerdos de Nivel de Servicio (SLAs).

Actualmente, los principales operadores de red y la comunidad científica se encuentran trabajando en estrategias para facilitar el análisis de datos y el proceso de toma de decisiones cuando eventos específicos comprometen la salud de las redes 5G. Al mismo tiempo, el concepto de Conciencia Situacional (SA) y los modelos de gestión de incidencias aplicados a redes 5G están en etapa temprana de desarrollo. La idea principal detrás de estos conceptos es prevenir o mitigar situaciones nocivas de manera reactiva y proactiva. En este contexto, el proyecto Self-Organized Network Management in Virtualized and Software Defined Networks (SELFNET) combina los conceptos de SDN, NFV and SON para proveer un marco de gestión autónomo e inteligente para redes 5G. SELFNET resuelve problemas comunes de red, mientras mejora la calidad de servicio (QoS) y la Calidad de Experiencia (QoE) de los usuarios finales.

Con este propósito, en esta tesis se presenta la aplicabilidad de las tres fases del procesamiento de información del modelo Situacional de Endsley. Este enfoque toma en cuenta los lineamientos de gestión tradicional y el dinamismo de los ambientes 5G para sentar las bases de un sistema con conciencia situacional. El modelo de conciencia situacional de SELFNET tiene el conocimiento de lo que está sucediendo con los diferentes elementos de la red, cómo optimizar su rendimiento y cómo solucionar o prevenir fallos.

Un pilar fundamental del enfoque presentado es el análisis contextual de la información recolectada procedente de los dispositivos de red, el cual es un tema emergente en este tipo de entornos. Por consiguiente, esta tesis también propone el Marco de Análisis de SELFNET para diagnosticar el estado de la red y predecir problemas potenciales, facilitando el proceso de toma de decisiones en entornos 5G. Esta propuesta proporciona capacidades para el reconocimiento de patrones, razonamiento y predicción con el objetivo de inferir conductas sospechosas y luego facilitar respuestas reactivas y proactivas. Además, el Marco de Análisis de SELFNET utiliza una metodología basada en casos de uso, donde el operador es capaz de personalizar los parámetros, las funciones de análisis y las reglas tomadas en cuenta en el proceso de diagnóstico. El resultado del Marco de Análisis de SELFNET es una solución escalable y simple capaz de satisfacer los requerimientos de las redes 5G. .

Palabras clave: Análisis de Datos, Conciencia Situacional, Gestión de Incidencias en 5G,

Redes Auto-organizadas, Redes Definidas por Software, SELFNET, Virtualización de las Funciones de Red.

Part I

State of the Art

Chapter 1

Introduction

The rapid proliferation of the use of mobile devices has revealed the lack of ability of the current networks to accommodate the vast amount of information that they will have to manage. This situation has given rise to the development of a brand new generation of mobile networks not only to provide solutions to such problems, but also to improve many features of their predecessors. Enhanced capabilities, related to transfer massive data, low latency, interoperability or reduction in energy consumption, allow for a better quality of experience to the end users. Achieving these goals requires great capacity for innovation, such as high speed data rates or better information management and analysis methods. The last part has significant impact on business models based on services and real-time applications (e-health, e-security, Voice over IP, streaming, etc.). However, at present, the development of these services is limited by the slow standardization process and the poor performance in the management and decision-making strategies. For this purpose, the fifth generation mobile network, or 5G, proposes the combination of some emerging technologies such as *Software Defined Networking* (SDN), *Network Function Virtualization* (NFV), cloud computing, *Self Organizing Networks* (SON), machine learning, artificial intelligence, among others.

SDN is based on the separation of the control plane from the data plane in traditional network devices. This decomposition allows the centralized control of the network with greater automation capacities and the simplification of management tasks, while accelerate the innovation of new high level applications. For its part, NFV enables the implementation of traditional *Network Function* (NF) as virtualized instances, running in a generic hardware. The scalable approach provided by NFV allows that *Virtual Network Functions* (VNFs) can be deployed anytime and anywhere in minutes, whereas previously it tooks more time (compared with traditional functions). From the technical point of view, SDN and NFV are complementary technologies, and together could facilitate configuration and network customization. Furthermore, concepts such as Cloud Computing, SON and Artificial Intelligence allow the easy deployment of services (on-demand fashion) and enhanced traffic management based on intelligent decisions.

On the other hand, there is a tendency to assume more cognitive methodologies in order to facilitate the environment understanding through contextual analysis such as the *Situational Awareness* (SA) model proposed by Endsley. In accordance with this method,

the perception, comprehension and projection of the system status must be taken into account in order to know what happen in the environment and how avoid or mitigate possible problems. In this context, the present research provides an Analyzer Framework able to diagnose different kind of problems in 5G Networks. This proposal is aware of the situational context of the elements and applications of this kind of environments, consequently provide intelligence capabilities.

The following sections summarize the research problem and the objectives of this thesis. Then, a short summary of the main contributions and the structure of this document are presented.

1.1 Research Problem

The exponential increase of on-line services (e-bank, e-health, streaming) and the number of connected devices has brought new challenges to the mobile network infrastructure in terms of security, performance and reliability. The management and rapid response to unexpected network problems (link failure, congestion, *Distributed Denial of Service* (DDoS), delay) is fundamental to guarantee *Quality of Service* (QoS)/*Quality of Experience* (QoE) to users, while decreasing the service recovery time and the capital and operational expenditures (capex and opex). On the one hand, the customization of network services requires the individual configuration of each device. On the other hand, the introduction of new solutions is limited by the rigidity of traditional network architectures due to their standardization process takes a long time (from design to implementation). For its part, data analysis and network intelligence mechanisms are needed in order to resolve or mitigate possible problems.

In order to aid in the resolution of these issues, the research community has proposed the integration of novel concepts like SDN, NFV, artificial intelligence, etc. Their combination provides a wide range of new lines of investigation, one of them the provisioning of data analysis and intelligence capabilities to 5G Networks. Nowadays, there are different projects that intend to meet intelligence and self-management requirements in 5G scenarios. However, it is important to note that the first advances on 5G are expected in 2020 and thus they are still work in progress. Because of that, the provisioning of data analysis and intelligent capabilities is the main research problem of this work.

Considering this opportunity, this work started with the idea of getting deeper in the subject of key-enabled technologies on 5G environments, their requirements, and how data analysis can take advantage of their capabilities. As a result the foundations of SELFNET Analyzer Component was laid.

This research is developed under Self-Organized Network Management in Virtualized and Software Defined Networks H2020 project (*Self-Organized Network Management in Virtualized and Software Defined Networks* (SELFNET)). SELFNET project aims to provide an autonomic network management framework for 5G mobile networks based on the combination of novel technologies such as SDN, NFV, SON, cloud computing and artificial intelligence. SELFNET enables the autonomic deployment of virtual network functions (sensors and actuators) and the reconfiguration of network parameters in order

to mitigate existing or potential problems, while maintaining the QoE of end users. These capabilities are provided by means of a layered architecture and a use-case driven approach, taking into account three main use cases: i) self-protection capabilities to mitigate or prevent security threats such as a DDoS or a cyber attack, i) self-healing capabilities to avoid or correct network outages/failures and iii) self-optimization to dynamically enhance the network and service performance. In order to achieve a high degree of automation and cover these use cases, data analysis and intelligence concepts are required by SELFNET.

In general terms, SELFNET and 5G systems require advanced capabilities to diagnose suspicious situations or unexpected changes within the network infrastructure. Detection of security threats, congestion forecasting, link failures, automatic deployment of network functions where they are required, shutdown devices to achieve network efficiency, etc, are examples where data analysis and intelligence are required. In SELFNET, these capabilities are intended to automate and enhance the performance not only of typical tasks but also of hazardous events. For this purpose, SELFNET intelligence is provided in two phases:

- Analysis task is intended to diagnose and infer the real network status based on monitored data from sensors and SELFNET data sources. Its main objective is to facilitate the decision-making process behind suspicious conditions.
- Then the decision-making task applies advanced intelligent techniques in order to determine the better action or countermeasure to be applied in the network, based on the symptoms provided by analysis task.

In this context, this research work proposes the SELFNET Analyzer framework to identify and analyse suspicious conditions based on metrics provided by sensors and different network devices. SELFNET Analyzer framework is aware of the situational context of 5G infrastructure and takes into account advanced techniques related to data analysis.

1.2 Objectives

Bearing in mind the dynamicity and intelligent needs of 5G networks, the main objective of the present research is to provide a situational awareness model for data analysis in 5G environments. This proposal pays special attention to data analysis, prediction, pattern recognition, adaptive threshold and knowledge inference capabilities. The key idea behind these actions is to facilitate the decision-making process in order to solve or mitigate common network problems in a reactive and proactive way. Taking into account the need to know the situational context in 5G networks, the following objectives have been conducted during this research:

1. Firstly, this research reviews the state of the art related to SDN, NFV and 5G in order to define what elements must be taken into account in the diagnosis process and how 5G key-enabled technologies aid in this purpose. Furthermore, the requirements and the advantages of incorporating advanced data analysis capabilities are discussed.

2. As a second step, the general description of SELFNET Framework is done in order to introduce the inputs/outputs and different requirements that SELFNET Analyzer Framework must consider.

3. Then, a situational awareness model for 5G network is proposed. This approach takes into account intelligence needs and provide a complete view of network status, thus facilitating the incident management and decision-making process.

4. Once the situational context and data analysis requirements are defined, the SELFNET Analyzer Framework is proposed. This framework creates a scalable and modular solution driven by use cases. SELFNET Analyzer Framework identifies suspicious or unexpected situations based on metrics provided by the 5G network components, the analysis rules and other parameters defined by use case operators.

1.3 Summary of the Contributions of this Thesis

The contributions of this thesis are organized in four main knowledge domains: i) Software Defined Networking (SDN) and Network Function Virtualization (NFV), ii) 5G Mobile Network (5G), iii) Situational Awareness and Incident Management and IV) Data Analysis. A short schema of the contribution of this thesis is shown in Figure 1.1. Bearing in mind the main objective of this work, that is the provisioning of a data

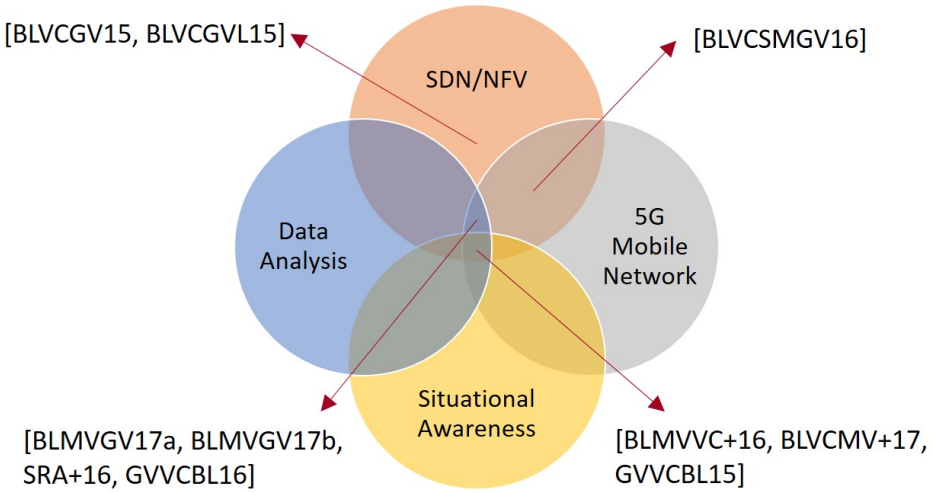

Figure 1.1: Contributions of this thesis.

analysis framework aware of the situational context of 5G networks, four main fields are studied. This thesis expects to provide advanced capabilities related to data analysis and self-management based on 5G key-enabled technologies. Regarding this, the contributions focused on SDN and NFV are presented in [BLVCGV15] and [BLVCGVL15]. Then, the contribution published on [BLVCSMGV16] contains the applicability of SDN/NFV on 5G environments and provides the current status of research work. Taking into account the above mentioned contributions, this thesis proposes a generalized approach to aid in incident management process through [BLMVVC+16] and [BLVCMV+17] contributions.

In turns, this approach defines an Analysis stage to know the real network status. For this purpose, the contribution of SELFNET Analyzer Framework and its components is done in [BLMVGV17a] and [BLMVGV17b]. To the best of our knowledge this is the first proposal that integrates both concepts: data analysis and situational context in 5G Networks. This is aligned to the capabilities of SELFNET framework [SRA+16], [GVVCBL15] and [GVVCBL16]. Consequently, these contributions cover all the topics analyzed on this work.

1.4 Outline of the Thesis

The thesis is organized in two main parts: State of the Art and Description of the Research. Part I reviews the advances on SDN, NFV, 5G and describes the SELFNET Project. This part consists of Chapters 2 to 6 respectively. In Part II the main contribution of this thesis are described: the framework for incident management in 5G Networks (Chapter 7) and the Analysis Module for 5G Infrastructures (Chapter 8). The detailed description of each Chapter is as follows:

In Chapter 1, the general information of this work is presented. It consists of the summary, the research problem, the objectives and the contributions within four different domains.

In Chapter 2, the state of the art related to Software Defined Networking is presented. This chapter reviews the evolution from traditional network environments towards SDN architectures. Then, the SDN architecture and its characteristics were described. It includes the description of its communication components such as OpenFlow protocol or SDN controller. This chapter also includes the ongoing work and its applicability in different fields.

In Chapter 3, the state of the art of Network Function Virtualization is introduced. The evolution of traditional network functions and the advances in virtualization are reviewed. Then, the NFV concept, its architecture and its components are described. It also includes the ongoing research and use cases on this field and its main implementation tools.

In Chapter 4, the vision of 5G Mobile Network is introduced. A general review of 5G capabilities and needs are presented. It takes into account the 5G requirements, its main *Key Performance Indicator* (KPI) and the fields that obtain advantage from 5G key enabled technologies.

In Chapter 5, the related works are presented. This chapter takes into account three main fields. Firstly, the research projects in 5G are presented. Second, the applicability of data analysis and diagnosis capabilities in these kind of systems are reviewed. Third, the incident information management applied to 5G Networks is analyzed.

In Chapter 6, the Self-organized Network Management in Virtualized and Software Defined Networks project (SELFNET) is introduced. This chapter describes the solution provided by SELFNET, its architecture and its components, laying the foundation of a self-management framework for SDN/NFV infrastructures.

In Chapter 7, a novel architecture for incident management on 5G networks is presented. This chapter describes the applicability of Endsley Situational Model in

conventional risk management schemes. The proposed approach takes into account the information from all layers defined in 5G networks.

In Chapter 8, the SELFNET Analyzer Framework is presented. This chapter describes how the proposed approach helps in the decision-making process. For this purpose, the design principles, data specification, high level architecture and its components are described. This chapter also provides information regarding how advanced data analysis techniques are used, such as prediction, pattern recognition and knowledge inference.

In Chapter 9, the conclusions and future work derived from this thesis are summarized.

1.5 Audience of this Thesis

The state of the art provides information and definitions in order to understand the main contribution of this research. A basic knowledge of mobile systems, protocols, incident management and data analysis are required and therefore the prerequisites to access this thesis material are not high. This work presents a situational awareness model for data analysis in 5G Networks and several references, which provides further information to be consulted by the reader.

Chapter 2

Software Defined Networking

This chapter reviews the main concepts related to SDN. Bearing in mind a broad view, the main applications and challenges are also presented and discussed. This chapter is organized in 6 sections. Section 2.1 reviews the traditional network architectures. Section 2.2 gives details of the separation of data and control planes proposed by SDN. Section 2.3 discusses the OpenFlow architecture. SDN Controllers and Network Operating Systems are the subject of Section 2.4 Section 2.5 presents a review of the SDN applications. Lastly, Section 2.6 summarizes this chapter.

2.1 Traditional Network Architectures

The idea of transmitting information between two points through a network led to the design of communication protocols (TCP/IP, HTTPS, and *Domain Name System* (DNS)) and the creation of specialized devices in the transmission of information. These devices have evolved resulting in a variety of equipment (hub, switch, router, firewall, *Intrusion Detection System* (IDS), middlebox, and filters). This development has produced an exponential increase in the number of connected devices, transmission rate and the emergence of online services (e-banking, e-commerce, e-mail, VoIP, etc.).

All devices responsible for transmitting information have similar features in their design and manufacture. First, there is a specialized hardware in the packet processing (data plane), and over the hardware works an operating system (usually Linux) that receives information from the hardware and runs a software application (control plane). The software contains thousands of lines of code for determining the next hop that a packet should be taken in order to reach its destination. The program follows the rules defined by a specific protocol (there are currently about 7000 *Request for Comments* (RFCs)) or some proprietary vendor technology. Modern equipment also analyzes information packets to search malicious information or intrusions (firewalls and IDS). However, all technology or software used in the manufacturing of these devices is rigid or closed to the network administrator.

The administrator is limited only to configure some parameters, usually through low level commands using a *Command Line Interface* (CLI). Moreover, each node is an autonomous system which finds the next hop to be taken by a packet to reach its

destination. Some protocols (*Open Shortest Path First* (OSPF), *Border Gateway Protocol* (BGP)) allow the nodes to share control information between them, but only with its immediate neighbours and in a limited way in order to avoid traditional load on network. This means that there is not a global view of the network as a whole. If the users need to control and modify a particular path, the administrator has to test with parameters, priorities, or uses gadgets to achieve the expected behaviour in the network. Each change in the network policy requires individual configuration directly or remotely from each of the devices. This rigidity makes the implementation of high-level network policies difficult.

Moreover, the policies are required to be adaptive and dynamically react according to the network conditions. As *Operating Systems* (OSs) evolve and adapt to the user needs and technological trends (support multi-CPU, multi-GPU, 3D, touch screen support, etc.), the network adaptability to new requirements (*Virtual Local Area Network* (VLAN), IPv6, QoS, and VoIP) is implemented through protocols or RFCs. However, in the operating system the separation between hardware and software allows the continuous update of application, or even the reinstallation of a new version of an OS. In the area of networks, the design and implementation period of a new idea could take several years until it is published in a protocol and incorporated in new devices. Some services are proprietary of the vendors and require that all network infrastructure belong to the same vendor to work properly. This limitation brings on the dependence on a specific technology or vendor.

2.2 Software Defined Networking

The concept of Software Defined Networking is not new and completely revolutionary; rather it arises as the result of contributions, ideas, and developments in research networking. In [Cal99], three important states are determined in the evolution of SDN: Active Networks (mid-90s to early 2000), separation of data and control planes (2001-2007), and the OpenFlow *Application Programming Interface* (API) and *Network Operating System* (NOS) (2007-2010). All these aspects are discussed below.

2.2.1 Active Networks

The difficulty for researchers to test new ideas in a real infrastructure and the time, effort and resources needed to standardize these ideas on the *Internet Engineering Task Force* (IETF) necessarily give some programmability to network devices. Active networks offer a programmable network interface or API that opens the individual resources of each node for the users, such as processing, memory resources, and packet processing and includes personalized features for the packets that circulate through the node. The need to use different programming models in the nodes was the first step for research in network virtualization, as well as the development of frameworks or platforms for the development of application on the node.

The Architectural Framework for Active Networks v1.0 [Cal99] contains a shared Node Operating System *Node Operating System* (NodeOS), a set of *Execution Environments* (EEs)), and (*Active Applications* (AAs)). The NodeOS manages the shared resources,

while the EE defines a virtual machine for the packet operations. The AA operates within an EE and provides the end-to-end service. The separation of packets to each EE depends on a pattern in the header of incoming packets to the node. This model was used in the PlanetLab [Pla17] platform, where researchers conducted experiments in virtual execution environments and packets were demultiplexed to each virtual environment based on its header. These developments were important, especially in the investigation of architectures, platforms, and programming models in networks. However, their applicability in industry was limited and mainly criticized for its limitations in performance and safety. The work presented in [WT01] is an effort to provide the best performance to the active networks, and the Secure Active Network Environment Architecture [AAKS98] tried to improve their security.

2.2.2 Separation of Data and Control Planes

The exponential growth in the volume of traffic over the network produces the necessity to improve the supervision process and uses best management functions such as the management of paths or links circulating the network (traffic engineering), prediction traffic, reaction, and fast recovery if there are network problems, among others. However, the development of these technologies has been strongly limited by the close connection between the hardware and software of networking devices. Besides, the continuous increase in link rates (backbones) means that the whole transmission mechanism of packets (packet forwarding) is focused on the hardware, separating control, or network management to an application of software. These applications work best on a server, because it has higher processing and memory resources compared with a single network device. In this sense, the project *Forwarding and Control Element Separation* (ForCES) [YDAG04] standardized by the IETF (RFC 3746) established an interface between data and control plane in the network nodes. The SoftRouter [LNR+04] used this software interface to install forwarding tables in the data plane of routers. Additionally, the *Routing Control Platform* (RCP) [CCF+05] project proposed logical centralized control of the network, thus facilitating the management, the innovation capacity, and programming of network. RCP had an immediate applicability because it uses an existing control protocol BGP to install entries in the routing tables of the routers. The separation of data plane and the control plane allows the development of "clean-slate" architectures, such as the 4D project [GHM+05] and Ethane [CFP+07]. 4D architecture proposes architecture of four layers based on functionality: data plane, discovery plane, dissemination plane, and decision plane. Moreover, the Ethane project [CFP+07] proposes a centralized control system of links to business networks. However, the need for custom switches based on Linux, OpenWrt or NetFPGA with support for Ethane protocol made the applicability of this project difficult.

2.2.3 SDN and OpenFlow

At the present time, the OpenFlow protocol [MAB+08] is the most widely used in the research community and it has been the basis of different projects. Companies

like Cisco have also submitted a proposal for a new architecture called Cisco Open Network Environment (Cisco ONE) [KK13]. Simplifying the previous analysis, the term Software-Defined Networking SDN proposes some changes to the networks of today. First, the separation or decoupling of the data plane and control plane allows for evolution and development independently. Secondly, it proposes a centralized control plane, thus having a global view of the network. Finally, SDN establishes open interfaces between the control plane and data plane. The differences between these architectures are shown in Figure 2.1.

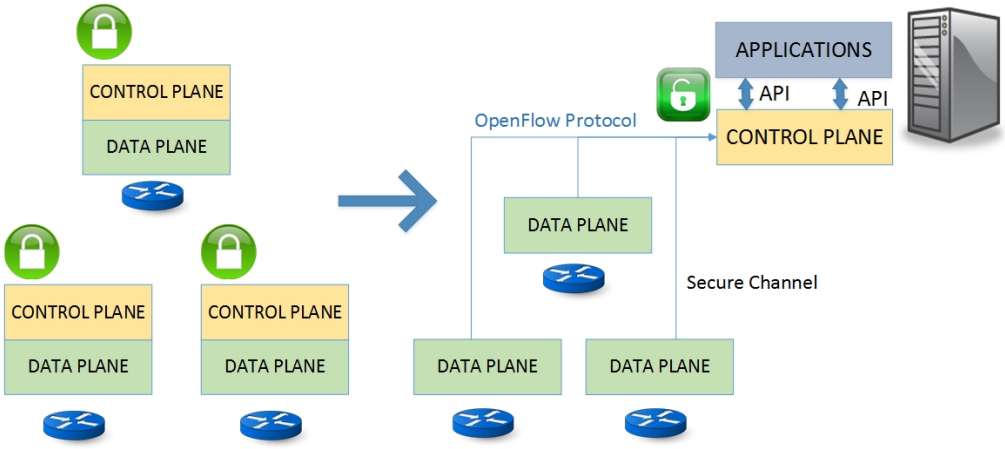

Figure 2.1: Comparison between traditional and SDN architectures

The programmability of the network provided by SDN can be compared with the mobile applications running on an Operating System (Android and Windows Mobile). These applications use the resources of the mobile (GPS, accelerometer, and memory) through the API provided by the OS. Likewise, the network administrator can manage and program resources in the network, according to user needs, through available APIs (proprietary or open) on the controller.

2.3 OpenFlow Protocol

OpenFlow [MAB+08] was originally proposed as an alternative for the development of experimental protocols on university campus, where it is possible to test new algorithms without disrupting or interfering with the normal operation of traffic of other users. Nowadays, the *Open Networking Foundation* (ONF) [ONF17] is the organization responsible for the publication of the OpenFlow protocol and other protocols for SDN, such as OF-Config [ope13].

The advantage of OpenFlow, compared with previous SDN protocols, is the use of elements and features of hardware available in most network devices. These elements are the routing tables and the common functions are as follows: read the header, send the packet to a port, and drop a packet, among others. OpenFlow opens up these elements and functions; so these can be controlled externally. This implies that, with a firmware update, the actual hardware could potentially support OpenFlow. The companies do not need a

complete change of their hardware to implement SDN in their products and services. The OpenFlow architecture proposes the existence of a controller, a switch OpenFlow, and a secure protocol of communication. These elements are shown in Figure 2.2. Each OpenFlow switch consists of flow tables that are managed by the controller. Each flow table has three elements: packet header, actions, and statistics. The packet header is like a mask that selects the packets which will be processed by the switch. The fields used for comparison can be from layer 2, 3, or 4 of the TCP/IP architecture. That means that there is not a separation between layers as in current architectures. All packets processed by the switch are filtered through this method. The number of fields that the switch can process depends on the version of the OpenFlow protocol. In OpenFlow v1.0 [ope09] (the most used version), there are 12 fields, while the latest version OpenFlow v1.3 defines the existence of 40 fields including support for IPv6.

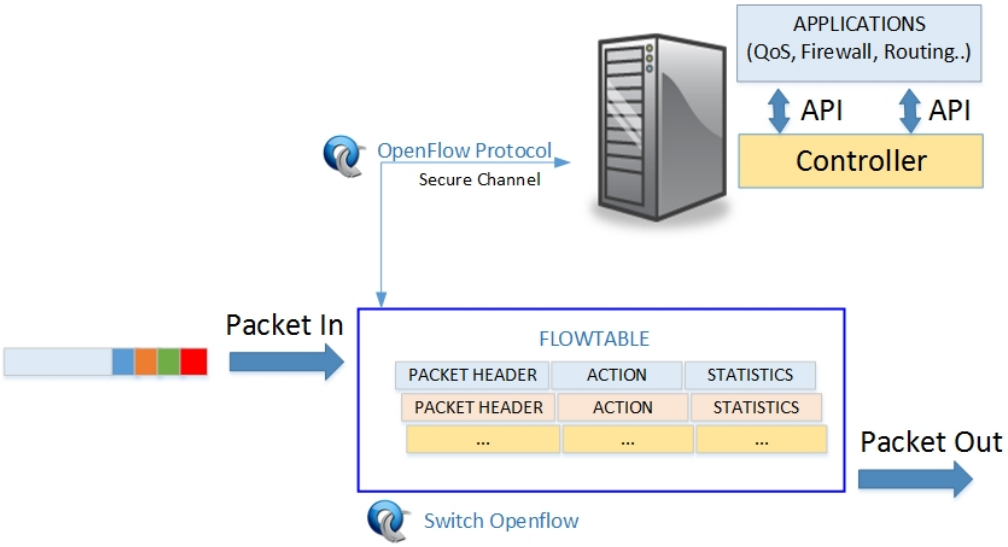

Figure 2.2: OpenFlow architecture

Once the header of an incoming packet matches the packet header of the flow table, the corresponding actions for that mask are performed by the switch. There are main and optional actions. The main actions are as follows: forward the packet to a particular port, encapsulate the packet and send it to the controller, and drop the packet. Some optional actions are as follows: forward a packet through a queue attached to a port (enqueue action) or 802.1D processing capabilities. If the header of an incoming packet does not match with the packet header of the flow table, the switch (according to its configuration) sends the packet to the controller for its analysis and treatment. Finally, the statistics field uses counters to collect statistic information for administration purposes.

An OpenFlow switch manages three kinds of table: flow, group and meter tables. The controller is able to add, delete or update flow entries in a flow table. In turns, each flow entry consists of the following elements: match fields (for matching packets), counters (tracking packets), priority, timeouts and a set of instructions. An OpenFlow switch uses a pipeline in order to define how the packets interact with flow tables, as is depicted in Figure 2.3.

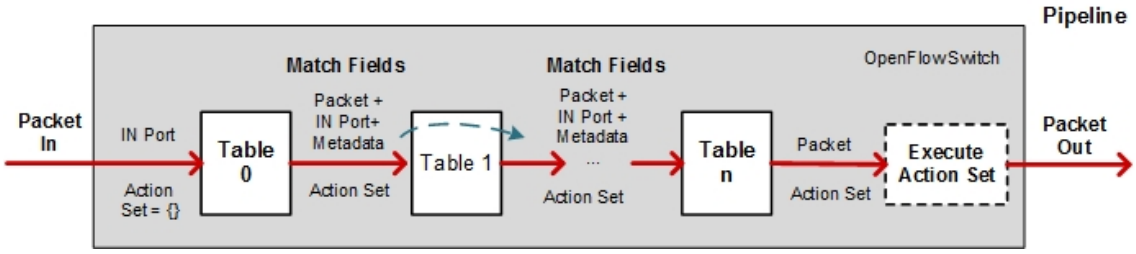

Figure 2.3: OpenFlow pipeline

The OpenFlow protocol defines the following types of messages between the switch and the controller: controller to switch, symmetric, and asynchronous. The messages type controller to switch manage the state of the switch. Symmetric messages are sent by the controller or switch to initiate the connection or interchange of messages. The asynchronous messages update the control of the network events and the changes of state switch. Similarly, OpenFlow establishes two types of switches: OpenFlow-only and OpenFlow-enabled. OpenFlow-only switches use only OpenFlow protocol to process packets. On the other side, OpenFlow-enabled switches can additionally process the packet using traditional algorithms of switching or routing.

The controller receives the information from the various switches and remotely configures the flow tables of the switch. Here, the user can literally program the behaviour of the network. Unlike active networks, which proposed a "Node Operating System" OpenFlow opens the notion of a NOS. In this respect, in [FRZ14], the NOS is defined as the software that abstracts the installation of the state in the switches of network of the logic and applications that control the behaviour of the network. In recent years, the NOS has evolved according to the needs and applications for researchers and network administrators.

2.4 SDN Controllers

The concept of NOS is based on the function of an operating system in computing. That is, the Operating System allows user to create applications using high-level abstraction of information, resources, and hardware. In SDN, some authors [SSHC+13], [RFR+12] have classified the abstractions of network resources as southbound and northbound interfaces (Figure 2.4). The function of the southbound interfaces is to abstract the functionality of the programmable switch and connect it to the controller software. A clear example of southbound interface is OpenFlow. On the southbound interfaces, you run a Network Operating Systems. An example of NOS is NOX [GKP+08], among others. On the other hand, the northbound interfaces allow applications or high level network policies to be easily created and they transmit these tasks to NOS. Examples of these interfaces are Frenetic [FHF+11], [FGR+13], Procera [KF13], [VKF12], Netcore [MFHW12], and McNettle [VW12].

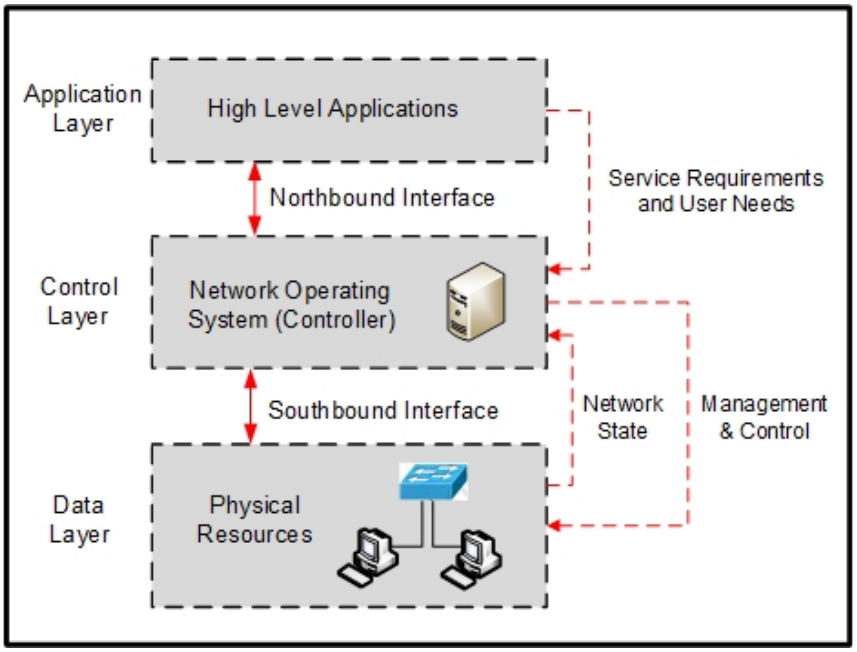

Figure 2.4: NOS southbound and northbound

2.4.1 Network Operating Systems

The NOX software [GKP+08] is the first NOS for OpenFlow and consists of 2 elements: processes of controller (controller) and a global view of the network (network view). Depending on the current state of the network, the user can make decisions and set the network behaviour through these processes. In NOX, traffic is handled at the level of flows (flow-based granularity); that is, all packets with the same header are treated similarly. The controller inserts, deletes entries, and reads the counters found in the flow tables of the switches. Furthermore, due to the dynamic nature of traffic, NOX uses events (event handlers) that are registered with different priorities to be executed when a specific event occurs in the network. The most used events are switch join, switch leave; packet received, and switch statistics received. Additionally, NOX includes "system libraries" implementations and common network services. Finally, NOX is implemented in C++ providing high performance. Moreover, there is an implementation entirely in Python denominated POX, which provides a more friendly developed language.

Floodlight (FDL) [flo17] is a Java-based OpenFlow controller. The core of FDL is an evolution of a previous OF-controller known as Beacon [Eri13]. FDL provides a simple and unrestricted interface; that is, the user can freely use the constructors available in Java (threads, timers, sockets, etc.). Furthermore, FDL is a NOS based on events; that is, the user sets the events that the controller listens to. The interaction with OpenFlow messages of the switch is done by the library OpenFlowJ, an implementation of the OpenFlow 1.0 [ope09] protocol and a Provider interface that contains the following listeners: IOFSwitchListener, IOFInitializerListener, and IOFMessageListener. Additionally, FDL has multithreading support and provides important APIs implementations (Device Manager, Topology, Routing, and Web UI) as well as the ability to start, add, and complete

applications without completely terminating a process in FDL (runtime modularity).

OpenDaylight (ODL) [MVTG14] is a NOS that uses *Model driven Software Engineering* (MDSE) and model driven network management in order to provide flexibility and scalability in the development of SDN applications. In this way, users can include multiple services and application using the ODL southbound plugins. Modeling language such as YANG together with NETCONF/RESTCONF protocols are used to facilitate developers the design of network services/applications. For its part, ONOS [BGH+14] is a NOS focused on providing a distributed SDN control platform. For this purpose, it proposes a distributed architecture in order to ensure availability and scale-out. Similarly, it provides a global network view and a logically centralized control even though the servers can be distributed across multiple locations.

2.4.2 High Level Network Policies Languages

Although a NOS can handle the flow tables of the switches, there are some problems that can cause malfunction of the network [SSHC+13], [RFR+12], [GRF13]. For example, the controller receives the first packet that arrives at the switch and has not matched a header in the flow table. Then the controller analyzes it, assigns actions, and forwards these instructions to the switch so that the other similar packages follow the same route. However, during this time, the second, third, or fourth similar packets can be received by the controller and cause an erratic operation. In other words, there are virtually two processes running, one on the controller and another on the switch, and these processes are not fully synchronized.

Another limitation is the composition; that is, if the user wants to configure two different services on the same switch (e.g., routing and monitoring), it is necessary to manually combine the two actions on the switch, prioritize, and keep the semantics of each element of the network. This makes the design, coordination, and reuse of the libraries very difficult. Additionally, the switch has to handle two types of messages simultaneously: packets and control messages. Any mismatch can cause a packet to be processed with an invalid policy and thereby causing major security problem on the network. For example, if there are two entries in a flow table with the same priority, the switch behaviour might be non-deterministic, because the execution would depend on the design of the switch hardware. For this reason, the research community has worked on secure interfaces that automatically interact and coordinate the correct behaviour of the switch (northbound).

Procera [KF13], [VKF12] is a framework that allows politics or high-level network configurations to be expressed. This architecture provides different actions and control domains to program the behaviour of the network. The main domains of control are as follows: time, data usage, flow, and status. With these domains, the user can determine a behaviour depending, for example, on the time of day, amount of data transmitted, privileges or groups of users, type of transmitted traffic, and so forth. Actions can be temporal or reactive and are expressed on a high-level language based on *Functional Reactive Programming* (FRP) and Haskell. In [VKF12] are the details of this language as well as examples of using Procera in monitoring applications and users control on a college campus. For its part, Frenetic [FHF+11], [FGR+13] is a high-level language dedicated to

SDN networks developed in Python. It is structured by 2 sublanguages: a Network Query Language and a Reactive Network Policy Management Library. The Network Query Language allows the user to read the status of the network.

This task is performed by installing rules (low-levels rules) on the switch which does not affect the normal operation of the network. In addition, the Network Policy Management Library is designed based on a language for robots, Yampa, [CNP03] and web programming libraries in Flapjax [MGB+09]. The actions use a constructor type rule containing a pattern or filters and action list as arguments. The main actions are as follows: sending to a particular port, sending packet to the controller, modification the packet header, and blank action that is interpreted as discard the packet. The installation of these policies is performed by generating policy events (queries), primitive events (Seconds, SwitchJoin SwitchExit, and PortChange), and listener (Print and Register). The results of experiments [FHF+11] shows that Frenetic provides simplicity and a significant savings in code and lower consumption of network resources compared to NOX.

One of the additional advantages of this language is the composition; that is, independent functional modules can be written and the runtime system coordinates its proper function in the controller and the switch. There are 2 types of composition: sequential and parallel. In sequential composition, the output of one module is the input of the next, for example, a load balancer that first modifies the IP destination of a packet and then searches the output port according to the new IP header. In parallel composition, both modules are executed virtually simultaneously in the controller; for example, if the balancer sends a packet with destination IP A to port 1, and packet B IP destined to port 2, this composition would result in a function that sends incoming packets for ports 1 and 2.

McNettle [VW12] is a controller specially designed to offer high scalability at the SDN network. This is achieved using a set of message handlers (one for each switch) having a function that handles the switch-local and network-state variables and manages the supply actions from the network flows. The idea is that the messages from the same switch are handled sequentially, while messages from different switches are handled concurrently. Similarly, each message is processed in a single core CPU to minimize the number of connections and synchronizations inter-cores among other performance improvements. The tests performed in [VW12] show that McNettle have a higher multicore performance compared to NOX or Beacon. The controller proposed in [GRF13] is based on the verification of the established politics, instead of searching bugs monitoring the controller operation. To perform the verification, the first step is to make use of the high-level language Netcore [MFHW12] to describe only the network behaviour. Then, theNetcore Compiler translates the politics to network configurations as flow table entries. The flow tables information is analyzed by the Verifier Run-time System which transforms the network configuration into a lower abstraction level named Featherweght OpenFlow. Featherweght OpenFlow is a model that use synchronization primitives to guarantee the coherent behaviour of the flowtables. Additionally, the Kinetic tool is described in [RFR+12]; this tool allows performing consistent updates in the network using two mechanisms: per-packet consistency and per-flow consistency. The per-packet consistency

mechanism ensures that a packet that is transmitted across the network is processed with the same configuration when an update occurs. The per-flow consistency mechanism ensures that every packet that belongs to the same flow (e.g., a TCP connection) will be processed in the same way by every switch in the network.

2.5 SDN Applications

Software-Defined Networking provides the ability to modify the network behaviour according to user needs. In other words, SDN itself does not solve any particular problem, but provides a more flexible tool to improve the network management. In order to test the advantages of this architecture, the research community has presented multiple projects of interest. Next, some of these applications are described.

2.5.1 Home Networking

In the emerging topic of *Internet of Thing* (IoT), the management of devices and network resources in home networks is a big challenge due to the number of users and devices connected to the same point (usually an access point). In [KF13], [KSC+11], the authors present an implementation of an OpenFlow-based system that allows the monitoring and management of user and control of the Internet access based on "usage caps" or a limited data capacity for each user or device. The system provides visibility of the network resources and management of access based on user, group, device, application, or time of day and even enables the ability to exchange data capacity with another user. The system of control and network monitoring uses the friendly interface Kermit. The capacity management and network policies are based on the Resonance language [NRFC09].

2.5.2 Security

The global vision of the network can improve the security of the systems. This security cannot be based only in the host-security, because such defenses are ineffective when the host is compromised. In [RMTF09], the Pedigree system is presented as an alternative to provide security in the traffic moving in an enterprise network. This OpenFlow-based system allows to the controller the analysis and the approval of connections and traffic flows in the network. The host has a security module in the kernel (tagger) that is not under users control. This module labels the connections request to send information through the network (processes, files, etc.). This label is sent to the controller (arbiter) in the start of the communication. The controller analyzes the tagger and accepts or rejects the connection according to its policies. Once the connection is authorized, the corresponding flow tables are installed in the switch. Pedigree increases the tolerance to a variety of attacks, such as polymorphic worms. The systems increase the load in the network traffic and the host. However, this load is not higher than common antivirus software.

2.5.3 Mobile Networks

The devices in the infrastructure on mobile carrier networks share similar limitations as computer networks. Likewise, the carrier networks execute standards and protocols, for example, the *Third Generation Partnership Project* (3GPP) Project as well as the private vendor implementations. At this point, the SDN paradigm and its flow-based model can be applied on this kind of infrastructure offering better tools. *Software-Defined Mobile Network* (SDMN) [PWH13] is an architecture that enables openness, innovation, and programmability to operators, without depending on exclusive vendors or *Over the Top* (OTT) service providers. This model consists of two elements: *MobileFlow Forwarding Engine* (MFFE) and the MobileFlow Controller *MobileFlow Controller* (MFC). MFFE is a simple and stable data plane and with high performance. It has a more complex structure than an OpenFlow switch, because it must support additional carrier functions, such as layer 3 tunneling (i.e., GTP-U and GRE), access network nodes functions, and flexible charging. The MFC is the high performance control plane, where the mobile networks applications can be developed. Additionally, MFC has 3GPP interfaces to interconnect with different *Mobile Management Entitys* (MMEs), *Serving Gateways* (SGWs), or *Packet Data Network Gateways* (PGWs).

2.5.4 Multimedia

The multiple online multimedia services, for example, the real time transmissions, require high levels of efficiency and availability of the network infrastructure. According to studies presented by CISCO, the IP video traffic will grow from 70% in 2015 to 82% by 2020 [zet16]. Moreover, in the last years, the concept of QoE [LCMP12] gained particular strength, which attempts to redefine the QoS considering the level of user acceptance to a particular service or multimedia application. Therefore, SDN allows the optimization of the multimedia management tasks. For example, in [KSKD+12] is improved the QoE through the path optimization. This architecture consists of two elements: the *QoS Matching and Optimization Function* (QMOF) that reads the different multimedia parameters and establishes the appropriate configuration for this path, and the *Path Assignment Function* (PAF) that regularly updates the network topology. In case of degradation of the quality on the links, the system automatically modifies the path parameters taking in count the priorities of the users. Similarly, the project OpenFlow-assisted *QoE Fairness Framework* (QFF) [GEB+13] analyses the traffic in the network and identifies the multimedia transmissions in order to optimize them in function of the terminal devices and the network requirements.

2.5.5 Reliability and Recovery

One of the most common problems in the traditional networks is the hardness to recover a link failure. The convergence time is affected by the limited information of the node to recalculate the route. In some cases, it is necessary the intervention of the network administrator to reestablish the network datapath. At this point, the global vision of SDN enables the customizing of recovery algorithms. [SSC+12] proposed an OpenFlow-based

system that uses the mechanism of restoration and protection to calculate an alternative path. In restoration mechanism, the controller looks for an alternative path when the fail signal is received. Meanwhile, in protection the system anticipates a failure and previously calculates an alternative path. Similar to a failure on switch or routers, the malfunction of the SDN controller (NOS failure, DDoS attack, and application error) can cause a collapse of the whole network. Therefore, the reliability of the network can be ensured through backup controllers. However, it is necessary to coordinate and update the information of control and configuration between principal and backup controllers. The CPRecovery [FBMP12] component is a primary backup mechanism that enables the replication of information between primary and backup controller. The system uses the replication phase to maintain the updated backup controller and the phase of recovery that starts the controller backup at the moment it detects a failure of the principal controller.

2.6 Summary

This chapter summarizes the SDN architecture. First, the limitations of traditional network architectures are analyzed. Then, the differences between traditional and SDN architectures are described. Next, OpenFlow as the most relevant SDN southbound protocol is reviewed. Finally, the principal Network Operating Systems and the relevant applications are presented.

Chapter 3

Network Function Virtualization

This chapter reviews the main concepts of NFV. This chapter is organized in 5 Sections. Section 3.1 reviews the virtualization concept in traditional architectures. Section 3.2 gives details on the NFV concept. Section 3.3 discusses the main NFV implementation tools. NFV applications and use cases is the subject of Section 3.4. Lastly, Section 3.5 summarizes this chapter.

3.1 Virtualization in Traditional Architectures

In traditional networks, the infrastructure is composed by a large number of network devices, each operating their own private software and highly dependent in proprietary hardware. For this reason, the design and installation of new services usually require the individual software updating or the replacement of hardware. This rigidity increases the installation and operational costs.

In order to optimize the reuse of the available resources, the idea of share the infrastructure between different users, each with their private logical isolated space have won the attention of companies and service providers. In this context, similar to computer virtualization layer, where a hardware abstraction permits slicing and sharing the hardware resources with different OS in a host, the goal of network virtualization is to isolate multiple logical networks, each of them with completely different addressing and forwarding mechanism, but sharing the same physical infrastructure. In other words, in network virtualization, it is intended that multiple virtual networks can operate on the same infrastructure, each with its own topology and routing logic.

One of the first approaches on network virtualization is the VLAN technologies [TFF+13]. In VLAN networks, different users can share network infrastructures. However, the separation is controlled only by the network administration and with limited parameters (port number) and just work with known network protocols. The network administrator assign a VLAN as a logical network formed by a group of hosts in a single broadcast domain. The broadcast domain and the corresponding logical topology is different from the physical network topology. VLANs help in managing and reconfiguring current networks. However, the number of VLANs is limited to 4094 due the rigidity of the protocol.

Another approach in the development of virtualization solutions is the *Virtual Private Network* (VPN) [CB10]. A VPN creates a secure tunnel to communicates multiple sites. The secure tunnel is carried over a public network and the nodes can be geographically separated. The type of VPNs depends on the layer of transportation to be protected. For instance, the layer 1 VPN corresponds to the circuit switching domain or layer 3 VPN over a L3 protocol such as IP. In addition, there are other solutions that provides virtualization capabilities. The Active Networks opens the remote control of the network device and enables some programmability resources in network devices. This remote control of the network functions can be used to filtering the traffic to different users and provide virtualization capabilities. Similarly, the OpenvSwitch [PPA+09] open source tools allows the creation of virtual environments in software switches. This virtual switch can be configured to provide private network domains to different virtual machines in the same hardware. Furthermore, device manufacturers also includes private virtualization solutions in the *Network Interface Card* (NIC).

3.2 Network Function Virtualization

As outlined before, the virtualization is not a new concept. The virtualization refers to the abstraction of the logical resources from the physical resources, creating multiple logical instances over the same physical infrastructure [JP13]. In this context, the virtualization has reached different technical domains: virtualization of operating systems, computer hardware platforms, storage capacities and networks. In the field of networking, the virtualization was originally understood as enabling simultaneously multiple *Virtual Networks* (VNs) over a physical network. However, the virtualization principles also has been extended to other functionalities of networks.

Nowadays, the telecom providers has several challenges and high costs in order to update or install new network functions or appliances. Often, the appliances (e.g. firewall, *Data Packet Inspection* (DPI)) are deployed in propietary hardware or private software, and consequently, cannot be reused or modified by other service providers. Moreover, the rigidity and complexity of network deployments reduce the customization capabilities of the services provided by operators.

In this context, the Network Function Virtualization proposes the transferring of the different NF (routing, firewall, deep packet inspection DPI, gateway) as virtual software-based applications executed in IT platforms (servers, switches and storage) [MSG+16]. This new vision of IT services provides a major flexibility and scalability, facilitates the development cycles and reduce costs [CDLL15]. Figure 3.1 describes the differences between NFV and traditional architectures.

3.2.1 ETSI-NFV Architecture

The Service Providers and the *European Telecommunications Standards Institute* (ETSI), that includes more than 28 network operators and 150 telecommunication enterprises, has released the first NFV specification [ETS13b]. The virtualization of NFs enables

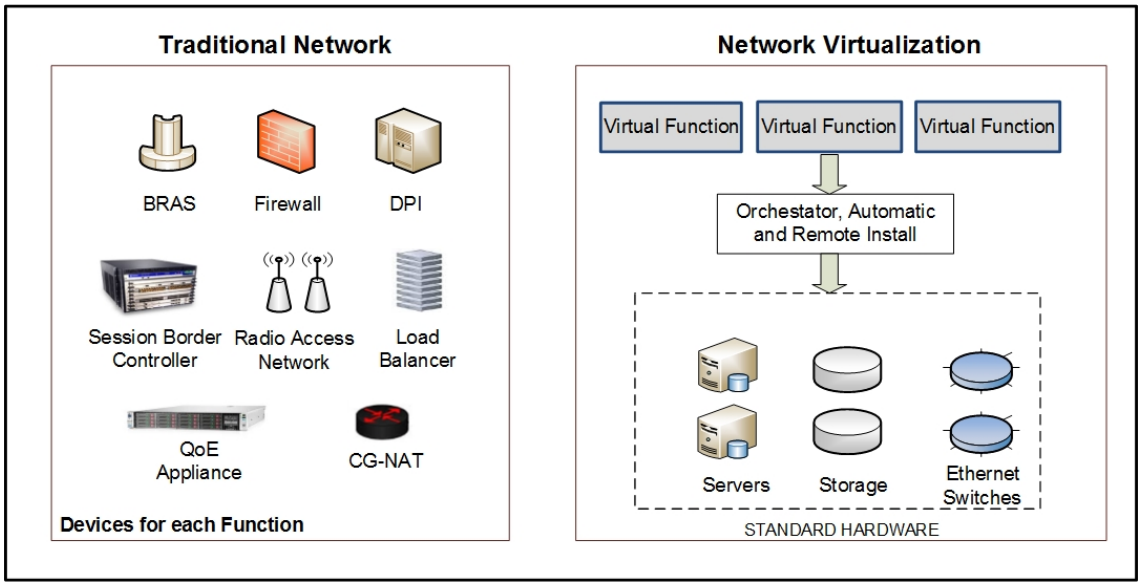

Figure 3.1: Traditional architectures vs. function virtualization

the running of virtual services on standard switches, storage or high-volume servers. Examples of NFs includes switching elements, mobile or sensor network nodes, traffic analysis functions, content distribution functions, *Session Border Controller* (SBC) and others. This new approach accelerates the deployment of new NFs, faster *Time to Market* (TTM), reduce of CapEx and OpEx and improves the customization and elasticity of services.

The NFV architecture released by ETSI is described in Figure 3.2 [ETS13a]. The NFV is composed of three main modules: *Network Function Virtualization Infrastructure* (NFVI), VNF and NFV Management and Orchestration (NFV M&O).

- **NFVI**. It represents the hardware and software resources of the system where the NFV concept are applied. Computing, storage and networking resources are included. The virtualization layers abstract the different resources and enable the isolation and independence of virtual compute, virtual storage and virtual network for different users.

- **VNF**. Represents an instance of a NF that runs over the NFVI. The VNF also includes the corresponding operational and management systems (*Operational Support System* (OSS) *Business Support System* (BSS)).

- **NFV M&O**. The principal function is the orchestration and management of the VNF and NFVI ensuring the optimal and effective operation of the Virtual Functions in the available infrastructure. The principal NFV M&O components are: the orchestrator, a VNF manager and the *Virtual Infrastructure Manager* (VIM). The VIM uses a resource inventory to control the availability of the different resources which guarantees the provisioning of services [ETS14].

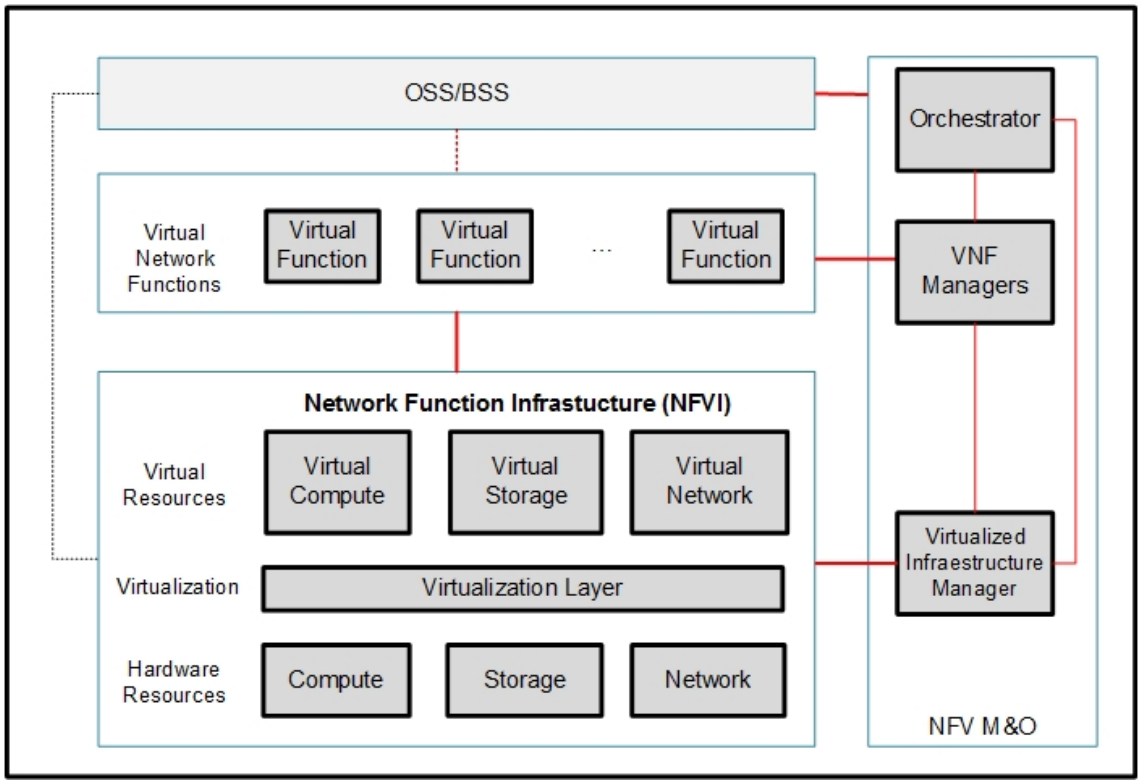

Figure 3.2: NFV architecture

3.2.2 IETF Service Function Chaining (SFC)

The *Service Function Chaining* (SFC) is defined as the interconnection of service instances, called service chaining, works by mapping packets to service chains at the edges and forwarding them between service instances [BRL+14]. In other words, SFC enables the organization and ordering of execution of multiples service functions that will be carried out by network devices during the transfer of information.

IETF establishes the IETF SFC Work Group (WG) which concentrates their efforts in the development of an open SFC architecture. IETF defines SFC as ordered set of abstract service functions and ordering constraints that must be applied to packets and/or frames and/or flows selected as a result of classification [HP15]. The IETF SFC architecture is depicted in the Figure 3.3.

The IETF SFC architecture [HP15] is composed of the following modules: *Service Function* (SF), Service Classification Function SCF, SFC-encapsulation and Service Function Forwarder.

- **SF**. It provides a specific processing in the received packets. The treatment can be executed at various ISO layers. The SF can be implemented as a virtual or physical network element. Similarly, a network element can provide multiple service functions and multiple SF occurrences can be executed in the same administrative domain.

- **SFC**. It receives the incoming traffic and splits them depending on a specific classification criteria. Its granularity depends on the SFC capabilities and the

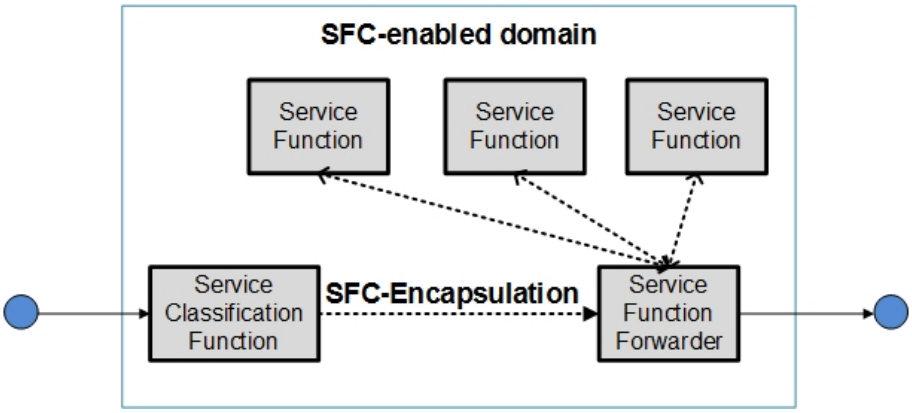

Figure 3.3: IETF SFC architecture [GMUJ16]

rigidity of the classification policies. Each of these traffic domain is known as Service Function Path SFP. Then, each SFP is sent to the SFC-encapsulation.

- **SFC-encapsulation**. SFC-encapsulation: It provides the required information to clearly identify a SFP during the function chaining process. The encapsulation information includes a SFP identification and the related metadata.
- **SFF**. It uses the information provided by the SFC-encapsulation to forward the traffic to the corresponding service functions. Then, once the traffic is processed, the SFF collects and transport the results of the SF to another SFF and finally terminates the SFP.

3.3 NFV Implementation Tools

The NFV principles completes the set of requirements to provide users a complete stack of cloud solutions. For this reason, open source and private cloud solutions have integrated NFV architecture in their products and tools [MVTG14], [KT217]. Although there is not a complete NFV open source tool, in this sections some advances on the implementation of different NFV elements are described.

3.3.1 OpenStack

OpenStack [ope17b] is the most widely open source tool used for cloud computing service development. Originally, OpenStack was developed by NASA and designed to provide an *Infrastructure as a Service* (IaaS) capabilities to service providers. However, its features have been extended to include the NFV capabilities in order to address additional advanced networking challenges.

The OpenStack architecture is depicted in Figure 3.4. The main components of OpenStack are: Compute, Networking and Controller.

- **Compute**. It is responsible of providing the requested compute instances. In this way, the OpenStack compute (also known as Nova) guarantees the spawning, scheduling and decommissioning of the requested virtual machines.

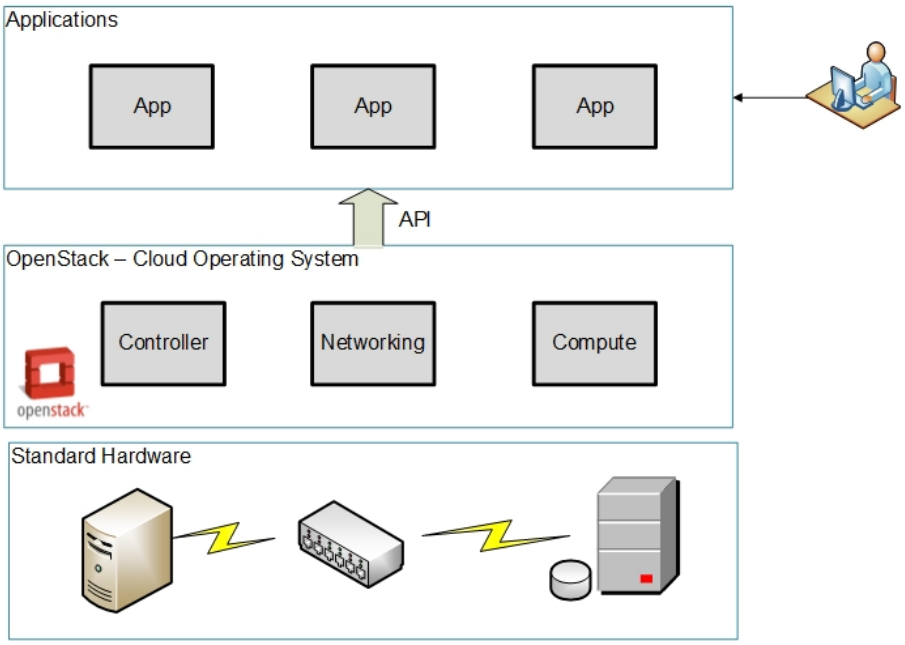

Figure 3.4: OpenStack architecture

- **Networking**. It provides the network connectivity (Network Connectivity as a Service) for the different OpenStack services. For instance, OpenStack Networking (also known as Neutron) enables the connectivity between the instances of compute VMs. In this way, the Neutron API allows the creation and management of the virtual networks and the architecture support multiple vendors and technologies.

- **Controller**. The controller includes different services that enables the provision of OpenStack services. For instance, Identity service, Image Service, Management elements of Compute and Networking or additional optional services. The number of controller components depends on the particular OpenStack deployment and requirements.

The relationship between OpenStack and NFV is that the services provided by OpenStack can be considered as the management of the virtual infrastructure (Virtual Infrastructure Manager) [JP13].

3.3.2 OpenBaton

OpenBaton [ope17a] is a ETSI NFV compliant *Network Function Virtualization Orchestrator* (NFVO). In other words, it coordinates the lifecycle of NFVs in the available virtual infrastructure. OpenBaton provides a Network Service Management using different Virtual Network Function Managers VNFMs. Similarly, its pluggable architecture enables the supporting of different VIM types, the easy integration with OpenStack and the integration with runtime management of Network Services.

The OpenBaton architecture is depicted in Figure 3.5. The principal components are: *Element Management System* (EMS), Virtual Network Function Manager VNFM, NFVO.

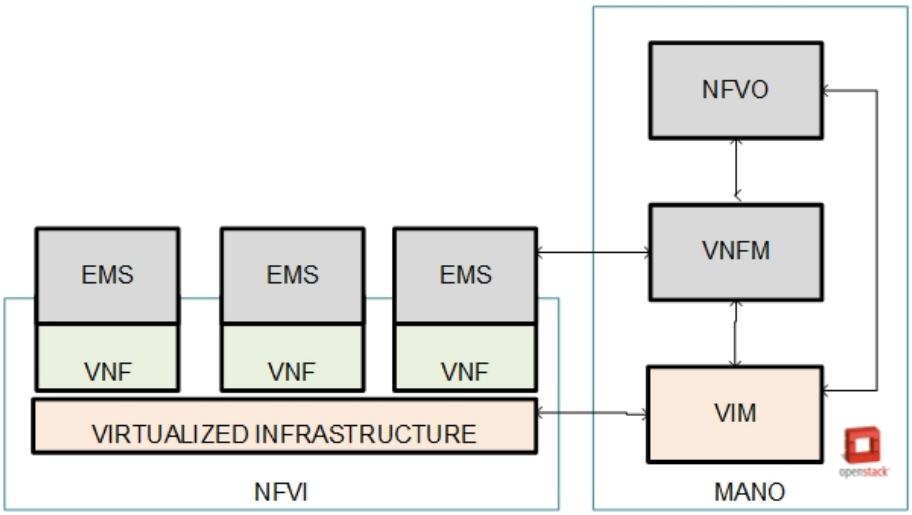

Figure 3.5: OpenBaton architecture

- **EMS**. It is an agent deployed inside the VMs that receives and processes the lifecycle events sent by the VNFM. For this purpose, it implements an *Advanced Message Queuing Protocol* (AMQP) producer/consumer. In this context, it subscribes to RabbitMQ message bus to receive the register-queue attending further commands from the VNFM.

- **VNFM**. It interoperates with the EMS in order to control the lifecycle of a VNF. The assigned management tasks can be realized in a single VNF instance or in a group of multiple and different types of VNFs. In this context, the generic VNFM communication is realized using AMQP over RabbitMQ.

- **NFVO**. It receives the request and controls the on-boarding process of a new network service or VNF instance in the virtualized infrastructure. The orchestration process includes the following tasks: NS lifecycle management, resource management, validation and authorization of NFVI requests, among others.

3.4 NFV Applications and Use Cases

The ETSI Industry Specification Group NFV-ISG [ETS] exposes a variety of NFV use cases. Depending on the application field, the NFV has several terminology definitions. The *NFV Infrastructure as a Service* (NFVIaaS) enables the deployment of NFs in the infrastructure of an external service providers. For its part, the VNF as a service (VNFaaS) allows the use or lease of a VNF to the infrastructure of private service providers. The VN platform as a service (VNPaaS) leases both the infrastructure and the available applications. The customers VNF can also be instantiated. The *VNF Forwarding Graph* (VNF-FG) facilitates the service chaining due the creation of a logical path with the hops NFs to deploy a service. Next, some of these applications are described.

3.4.1 Mobile Network Virtualization

The growing number of mobile devices and the increasing demand of high data rates and slow latency have pressured service operator to continually upgrade and enhance their infrastructure. In this context, the possibility of virtualizing the different functions and operations of a mobile infrastructure is gaining the attention of the research community [AHGZ16]. In [HSMA14], the authors faces the technical challenges and advances on virtualization of mobile networks. Similarly, they proposes an approach to the separation of data and control planes in mobile entities. In this way, the virtualization of an eNodeB (*Long Term Evolution* (LTE) base station) could reduce the number of operations required to connect the eNodeB to the mobile core network. Moreover, a virtual eNode could offer a centralized computing infrastructure for multiple base stations enabling the sharing of resources between different service providers. For its part, the virtualization of the mobile core network *Evolved Packet Core* (EPC) has the potential to reduce the opex and capex. An NFV-based EPC infrastructure could be scaled on-demand in real time and can be easily updated to support a variety of access technologies.

3.4.2 Optical Networks

The management of optical transport networks has evolved from a *Dense Wavelength Division Multiplexing* (DWDM) to dynamic switching technologies and flexible grid transmission schemes. In order to achieve this objective, the traditional *Network Management System* (NMS) for transport networks is improved by the flexibility provided by NFV approach. In [KFG15], the authors proposes a novel framework for the customized control of resources with a given user bandwidth demand. It uses flexi-grid, SDN/NFV and *Application Based Network Operation* (ABNO) to provide resilient and elastic network capabilities depending of actual and predicted demands. The architecture is composed of Network Control and NFV Management. The Network Control uses ABNO principles to control the optimal network while the NFV Management coordinate the availability of virtual resources. In this way, the users can develop end to end services ensuring enough frequency slot widths. The feasibility of the framework is demonstrated with the implementation of a virtual Content Delivery Network (vCDN) for video streams and TV services.

3.4.3 Network Virtualization Services

The NFV approach offers several opportunities to service providers in the development of new services and applications [KT417c], [tno17]. However, depending on the application, some NFV proposals can offer special advantages to developers in a specific area. Next, some additional NFV-based solutions for NF developers are described.

CloudNaaS [BASS11] is a novel framework that enables service providers the use of NFs and provide services in the infrastructure, such as isolation, QoS, custom addressing, among others. The design includes two main components: the cloud and network controller. The cloud controller coordinates the physical hosts and the corresponding virtual resources. For its part, the network controller is responsible for the virtual and

physical network devices. Similarly, CloudNaaS defines four principal operations. The first operation is the definition of requirements using a defined policy language. Then, the policies are translated to a communication matrix in order to find the optimal place to locate *Virtual Machines* (VMs). Then, the corresponding configuration rules are defined using a network language and, finally, the rules are installed in the network devices and VMs.

For its part, OpenADN [PJ12] proposes the use of SDN principles together with features of application to enable the provisioning of services and application in a distributed environment. The design is focused on cloud environments and includes four layers: virtualization layer to slice the network resources, NOS layer to control the OpenFlow network devices, a network level which invokes the ISPs services and a control level for the control of each application. The implementation includes technologies such as OpenFlow, MPLS, slicing and cross-layer communication.

The Project VMware NSX [vS13] enables the onboarding of a VN as fast as a VM in the provisioning of network virtualization services (e.g. compute or storage). For this purpose, NSX uses the Nicira Network Virtualization Platform (NVP) to provide programability to virtual networks using SDN and adding a layer between the network and final hosts. The layers of this architecture is described as follows. The data plane includes a virtual switch (NSX vSwitch) which abstract the physical resources and connect with hypervisor. In this scenario, the NSX Edge acts as a gateway between logical and physical network. Then, the NSX controller provides the control plane or SDN controller. Finally, the management plane implements a vSphere to enable the high level configuration and a cloud migration portal to control the migration and management tasks in virtual and cloud environments.

3.5 Summary

This chapter summarizes the NFV architecture. First, the virtualization in traditional architectures are analyzed. Then, the NFV principles are reviewed. Next, the NFV main implementation tools are reviewed. Similarly, the relevant NFV applications and use cases are presented

Chapter 4

5G Generation Mobile Network

This chapter reviews the main concepts related to 5G Networks, their requirements, ongoing work, (KPIs), future trends and challenges and key-enabled technologies that leverage these kind of systems. This chapter is organized in 5 sections. Section 4.1 presents a general overview of 5G networks. Section 4.2 describes the 5G requirements and the fields where they can be applied. Section 4.3 discusses the different Key Performance Indicators to measure the accomplishment of 5G requirements. In Section 4.4 the future trends and challenges are discussed. Lastly, Section 4.5 summarizes this chapter.

4.1 Overview

The emergence of a new business model and services (e-solutions, e-health, e-commerce, Voice IP, streaming, among others) and the exponential growth in the information circulating on the Internet has brought unexpected challenges to the IT industry. The development of new mobile infrastructures, is focused on ensuring robustness, security, scalability and the fast deployment of applications through the customization of network behaviour. According to the Future Internet 2020 Report of the European Commission, the development of a new generation of networks takes an average of 10 years, this means that the fifth Generation Mobile Systems (5G) are coming soon [HNS+09].

5G must provide a flexible, reliable, secure, smart and high-performance environment to connect the digital society, while leveraging the competitiveness, faster innovation and standardization of new technologies. This network must embrace not only current services but also any kind of elements (IoT). These kind of networks will generate a significant impact not only on the societal but also on the operational field. On one hand, 5G must cover the necessities of smart cities, entertainment, public security, etc., providing a wide range of network services and applications [Nok14]. Users will expect enhanced QoE with minimal disruptions of the services, regardless of their location, the kind of device, or when the service is required. On the other hand, 5G will help to decrease the capital and operational expenditures (capex/opex) related to the deployment and management of new applications and infrastructures with substantially reduced service creation time [AIS+14].

Nowadays, the introduction of novel technologies is a time-consuming process due

to the slow standardization process, manual service deployment or the semi-automated management tasks. At the same time, the *Average Revenue Per User* (ARPU) is continuously decreasing, while the demand on mobile traffic keeps growing. This causes a negative response by network operators to invest in new network hardware infrastructure. In order to lay the foundations of 5G Networks, three fields must be improved: Radio, Network and Operations and Management capabilities [EHE15].

- Radio Access capabilities leverage the spectrum optimization, enhance interference coordination mechanisms and support dynamic radio topologies through the exploitation of higher frequencies, enabling cost-effective dense deployments, intelligent and dynamic coordination of multi *Radio Access Technology* (RAT), as well as sharing resources, among others. Radio capabilities are intended to enable high data volumes, high mobility and spectrum efficiency.

- Network functionalities will enable the creation of an open environment in order to support several use cases in a cost-effective manner by means of the enhancement of user devices, minimizing the number of deployed entities and splitting the control and user plane functions (open its interfaces). These functionalities are also intended to ensure QoS levels.

- The operation and management capabilities are intended to simplify operations not only in network control tasks but also in the deployment of new services, without increasing the system complexity. This field also includes reactive and proactive mechanisms to enhance the decision-making in control and management operations based on network status and user profiles. This characteristic will enable the suited allocation of virtualized components, wherever they might be needed according the network status.

Radio and Network capabilities are topics well-studied in the literature [BHL+14], [GJ15], [BTAS14], [ABC+14]. In [GJ15], a detailed survey and ongoing projects related to 5G networks are presented. This work discusses some emerging technologies, such as massive *Multiple Input Multiple Output* (MIMO), cognitive radio, cloud technologies and *Device to Device Communication* (D2D) in order to tackle the following requirements: enhanced data rate, spectral efficiency, lower latency, deployment and management of ultra-dense networks. For their part, Boccardi et al. [BHL+14] describe five disruptive concepts that might impact on the development of 5G Radio requirements. They take into account the ability for devices to communicate between themselves (*Machine to Machine Communication* (M2M)), spectrum and resource optimization (massive MIMO and millimeter wave), the introduction of a device-centric concept and smarter devices (allowed to play an active role in the network). Regarding Network capabilities, one of the main challenges is to create an open, multi-tenant and service-oriented environment to support large amounts of traffic while covering different kinds of QoS levels and *Service Level Agreement* (SLA), in terms of latency, bandwidth or jitter. This environment will allow a flexible reconfiguration of network devices and programmability features based on the device-level, application, user and environment context [BTAS14]. Meanwhile, the

introduction of intelligence in 5G systems might enable the improvement of the resource use (spectrum, transmission power levels and other radio resources), cost-effective energy mechanisms and flexible cell management (different sizes) [DGK+13].

In order to tackle operation and management capabilities and enable ubiquitous connectivity, the research community proposes the introduction of some key technologies, such as SDN [KRV+15], NFV [ETS13a] Cloud Computing [ZCB10], Self-Organizing Network (SON) [BGMB14] and Machine Learning [BAX+16]. SDN is based on the separation of the control plane from the data plane in traditional network devices. This decomposition allows the centralized control of the network with greater automation capacities and it simplifies the management process. For its part, NFV allows the implementation of traditional NF as virtualized instances, running in a generic hardware. The main advantage of NFV is its improved scalability capacity which, due to VNF, can be deployed anytime and anywhere in minutes, whereas previously it took more time (compared with traditional functions) [MSG+16]. From the technical point of view, SDN and NFV are complementary technologies, and together could facilitate configuration and network customization [CDLL15]. For their part, concepts such as Cloud Computing and SON allow the easy deployment of services (on-demand fashion) and enhanced traffic management based on intelligence decisions.

SDN, NFV, Cloud computing and SON are enablers that provide business agility and simplify the operation and management tasks. In contrast to traditional mobile systems, future networks will enable operators to control the traffic information (via SDN) in order to use only necessary network functions in a shared virtualized network (NFV and cloud computing). These technologies also allow the reduction of the complexity of planning, configuration and optimization tasks in the whole system, giving the capability to reuse existing infrastructures in a proactive way.

For its part, steps have been made by some stakeholders to cover future 5G needs [Nok14], [EHE15], [Moh15], [SC15], [Net14], [NEC15] such the definition of requirements and use cases, standardization and regulation activities, 5G testbeds, definition of KPIs, among others. In these activities have participated not only the academy and the main telecommunication service providers but also regulation bodies, *Small Medium Enterprises* (SMEs) and *Standards Developing Organizations* (SDOs). These organizations have helped to define a set of 5G requirements to address the needs of future mobile users, as is explained in the following sections.

4.2 5G Requirements

5G, moves towards bringing solutions to deploying faster networks, with hundreds of thousands of simultaneous connections and massive data transfer, while ensuring the quality of the new services. For this purpose, some requirements have been defined in six main dimensions [EHE15], as is shown in Figure 4.1.

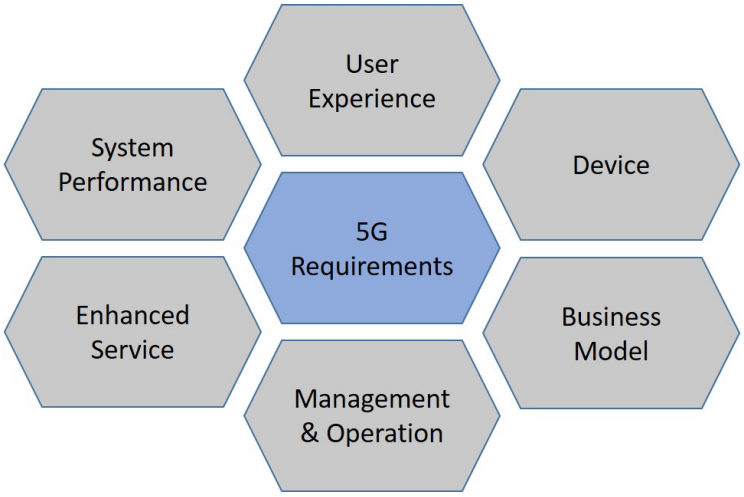

Figure 4.1: 5G requirements [EHE15]

4.2.1 User Experience

User Experience requirement focuses on maintaining and improving the user experience and their QoS levels across a broad range of mobility scenarios, in terms of bandwidth and delay. 5G users will benefit from consistent and high-quality service, regardless of network operator or their location with enhanced user experience compared with 4G systems even in critical situations such as very crowded areas, emergency events, network failures, etc. User experience includes high quality video services at HD or U-HD resolution [Nok14], [Net14] anytime and anywhere, enjoying a seamless experience.

4.2.2 System Performance

Currently, the spread of new services and devices, such as IoTs, autonomic vehicles, smart robots, virtual reality, etc, brings new requirements on system performance. 5G will be able to support the number of simultaneous connections per square kilometre, ultra-reliability, low end-to-end latency, which go beyond the traditional performance metrics, like capacity (data rate), coverage, the maximal mobility speed, and the number of active terminals per cell. As an example, the requirement on data rate will be substantially increased with capacities around 10 Gbps [Nok14], [EHE15], [Moh15], [SC15].

4.2.3 Devices Requirements

The main objective is to achieve high data rates, resources efficiency and signalling efficiency. New generation of devices like smartphones, robots, sensors, etc, will not only bring an increment of $x1000$ times in mobile traffic by 2020, but also impose a wide range of differentiated QoS requirements that will be met by 5G systems [Moh15]. Device to device D2D communication, aggregation capabilities and smart devices with multiple bands and multiple modes will be also required in 5G infrastructures. Another important issue is to increase the battery life of current devices, for both smartphones and low cost sensors [EHE15]. For this purpose, 5G terminals must have a high degree of programmability and

configurability for both hardware and software components, in terms of transport protocol, terminal capabilities, access technology, etc. The operator will be able to monitor and analyse the network and service information in order to detect possible problems such as video drops or link failure, and then optimize and prevent harmful situations.

4.2.4 Enhanced Services

This requirement is intended to improve the user experience through the transparent connectivity, advanced location capabilities, high availability, reliability and resilience while providing high level of security in 5G environments. End devices will be able to connect to several RAT in order to enable transparent connectivity [CZA+15]. It takes into account the connection with a specific RAT or a combination of RATs and a seamless connection from both the user and the network perspective. For its part, 5G systems will be aware of contextual information such as the location in order to provide tracking of moving terminals. 5G will also enable extremely high network availability and reliability and self-healing capabilities to improve network resilience, especially in critical situations such as public safety or natural disasters. Furthermore, 5G devices will enjoy enhanced security services provided by robust authentication and user privacy.

4.2.5 New Business Models

In contrast to traditional mobile networks, 5G will enable operators to configure the data plane of network devices in order to control and customize the network behaviour without having architectural impact. In this way, the network and service provider will be able to introduce new business models while reducing the capital and operational expenditures. Thus, 5G may accelerate the TTM of new services and create independence from the hardware vendor [Net14]. 5G will also facilitate the evolution towards supporting enhanced levels of abstractions based on the separation between control and data plane. This approach enhances the coordination and isolation of access, configuration and management capabilities between service and network providers, similar to current services such as *Platform as a Service* (PaaS) or *Network as a Service* (NaaS). Furthermore, the 5G system may provide advanced network sharing schemes in order to enable flexibility in the provisioning of services. This can include spectrum sharing or spectrum selection [SC15].

4.2.6 Deployment, Operation and Management

The main idea behind this requirement is to facilitate the provisioning and control tasks when a service is required, while reducing the capital and operational cost and to accelerate the TTM of new services. 5G systems enable a cost efficiency approach to minimize the *Total Cost of Ownership* (TCO) not only in the deployment of 5G infrastructures but also in management tasks. In general terms, 5G systems will provide self-configuration, self-optimization and self-healing capabilities in order to reply to failures or unexpected problems. For this purpose, 5G networks shall be able to take decisions in a short time and then apply countermeasures. For instance, configuration changes or deployment

of new virtual functions [NCC+16]. In this way, 5G foster the resource and operation efficiency and the seamless innovation or upgrade. 5G will reduce complexity of planning, configuration and optimization tasks, giving the capability to reuse and smoothly upgrade existing network infrastructures. 5G design [Nok14], [Moh15] should provide reliability, not only on equipment uptime, but also in the provision of required data, regardless the specific technology or vendor. This characteristic is crucial on mobile communications for control and mission critical services. Additionally, 5G is expected to cover areas with low population density, where ultra-low cost deployments are required due to the very low ARPU.

4.3 5G Key Performance Indicators

It is expected that 5G requirements will be covered in 2020 as well as beyond 2020. In order to define a performance measurement that reflect the accomplishment of 5G requirements, several KPIs are defined (Figure 4.2). These KPIs take into account the vision of different organizations and their main objective is to improve current capacities such as lower latency, more capacity and mobility, higher reliability and availability [Nok14], [AIS+14], [EHE15], [Moh15], [SC15], [Net14], [NEC15]. KPIs have a significant impact on 5G infrastructures in three levels: societal, operational and innovation.

Regarding the societal level, 5G is expected to enable ubiquitous, robust and continuous service access for end users. 5G will also be able to provide reactive and proactive responses against network problems. For this purpose, low latency and advanced management capabilities are required. For instance, the introduction of intelligent mechanism aids to decide if specific physical device is not been used and therefore it will be shut down in order to reduce energy consumption. At operational level, the main purpose of 5G systems is to decrease the capital and operational costs when new service is required. 5G will reduce the creation and deployment lifecycle of new services. With regard to innovation level, 5G fosters a wide range of opportunities in low dense areas, higher resource efficiency, advanced security, flexible transport network, extreme-reliable communications, among others [Moh15], [Net14].

In general terms, eight KPIs are common between the research community and several organizations: latency, peak data rate, mobility, number of connected devices, capacity, energy efficiency, location accuracy and operational cost (opex). The main idea behind these KPIs is to enhance the capabilities in each field, as is explained below.

- Peak Data Rate (10 Gbps)
  5G Networks will provide higher data rate than its predecessor LTE. It is expected to reach a peak data rate around $10 Gbps$ [Moh15], [SC15] regardless the user location or the number of connected devices.

- Latency (5 ms)
  5G expects to decrease the latency perceived by the user from the source to the destination (end-to-end latency). 5G also introduces the term “zero latency” which means there are no interruptions perceived by the user, taking into account the user

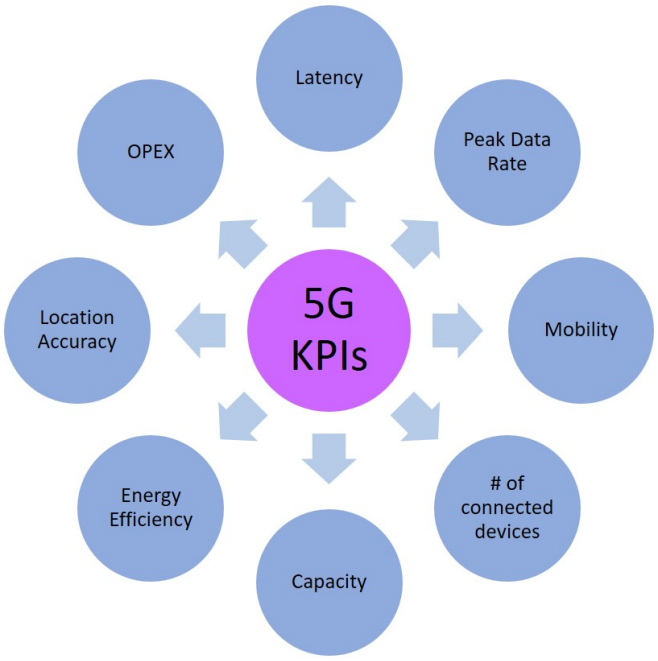

Figure 4.2: Summary of 5G key performance indicators

expectations, the service quality and continuity. This KPI is a critical parameter in real time environments, such as public safety, real emergency systems, augmented reality, among others [OBB+14].

- Number of Connected devices (1 M/Km2)
  5G will be able to support 1000x number of connected devices compared to current infrastructures. 5G will take into account the concept of sensor networks or IoT, which envisage increasing the number of simultaneous connections in mobile networks and is expected to cover around 1M (devices) per square kilometre.

- Mobility (500Km/h)
  This KPI enables end users to enjoy a seamless experience in fast-moving events like a train journey. This KPI is based on 5G user characteristics and their environment such as the diversity of mobile devices or the density of devices per square kilometre. 5G will also be able to support mobility speeds over 300 or 500km/h [SC15]. These capabilities also require higher reliability and lower latency depends on the environment for instance broadband in dense areas, broadband to vehicles, etc [EHE15].

- Capacity (10 Tbps/Km2)
  5G will provide access to different kind of activities where high capacities are required such as dense areas with thousands of users are connected. 5G systems will be able to handle higher volume of traffic and variations, regardless of the connection density [EHE15].

- Energy efficiency (10%)
  This KPI will cope the energy efficiency requirement on 5G networks. It is expected

that energy consumption per service decreases by 10% compared to 2010 [Moh15]. It includes not only end devices but also the whole network components such as cloud, *Radio Access Network* (RAN) and core elements [EHE15].

- Location accuracy (1 m)
  5G systems will be more efficient in terms of location accuracy (approximately 1m). This factor gains importance to deliver personalized services in real time services such as logistic transportation systems or road congestion information [NEC15].

- Service Creation Time (20%)
  This KPI is directly related to the reduction of operational expenditures (opex). 5G expect to reduce opex by 20% compared with current deployment, it means an approximate reduction of service creation time from 90 days to 90 minutes [Moh15].

Another transversal 5G KPIs are security robust, ubiquitous 5G access including in low dense areas, evolution of battery technology, reliability, improvements to facilitate dense deployments, among others [Nok14], [AIS+14], [EHE15], [Moh15], [SC15], [Net14], [NEC15]. In Table 4.1 the summary of 5G KPIs is presented, with their description and the expected value or percentage that they will meet.

Table 4.1: Summary of 5G key performance indicators

KPI	Description	Expected Value / Percentage	References
Latency	$\frac{1}{2}x$ end to end latency	$\leq 5ms$	[Nok14], [AIS+14], [EHE15], [Moh15], [SC15], [NEC15]
Peak Data Rate	$100x$ peak data rate	$\geq 10Gbps$	[Nok14], [AIS+14], [EHE15], [Moh15], [SC15], [NEC15]
Mobility	Mobility support to transport (vehicles)	$\geq 500Km/h$	[EHE15], [Moh15], [SC15]
Number of connected devices	$1000x$ number of connected devices	$\geq 1Mdevices/Km2$	[Nok14], [AIS+14], [EHE15], [Moh15], [SC15]
Capacity	Data volume density	$\geq 10Tbps/Km2$	[AIS+14], [EHE15], [Moh15], [NEC15]
Energy efficiency	10% lower energy consumption compared to traditional mobile networks (reduce $\frac{1}{10}x$)	10%	[AIS+14], [EHE15], [Moh15], [NEC15]
Location accuracy	Accuracy to determine the location of end devices	$\leq 1m$	[Moh15], [NEC15]
Opex	20% lower operational cost compared to traditional mobile networks (reduce $\frac{1}{6}x$)	20%	[AIS+14], [Moh15]

4.4 Future Trends and Challenges of 5G Networks

The current necessities address the direction of the business and the requirements of 5G Networks. It is expected that 5G networks will cover the increase of traffic volume by

means of improving spectrum utilization, enhanced energy efficiency mechanisms, resource virtualization, resource sharing, self-management and self-organization capabilities [ABC+14]. The concept of 5G envisages a broad range of opportunities in different fields. In other words, it will cover not only the traditional network fields but also other domains, such as e-health, energy efficiency, emergency services, public safety, IoT, M2M communication, *Information Centric Networking* (ICN), among others.

The applicability of SDN, NFV, SON and cloud computing opens the door to facilitate the deployment and management of services in an open business environment, as is shown in Figure 4.3. On one hand, it presents a layered structure: infrastructure, virtualization, control and application layers, similar to the SDN approach. On the other hand, VNFs and NFV M&O modules are incorporated in order to control the NFVI. For its part, cloud technologies are present on the cloud computing layer and SON capacities will aid in the decision process in the control layer.

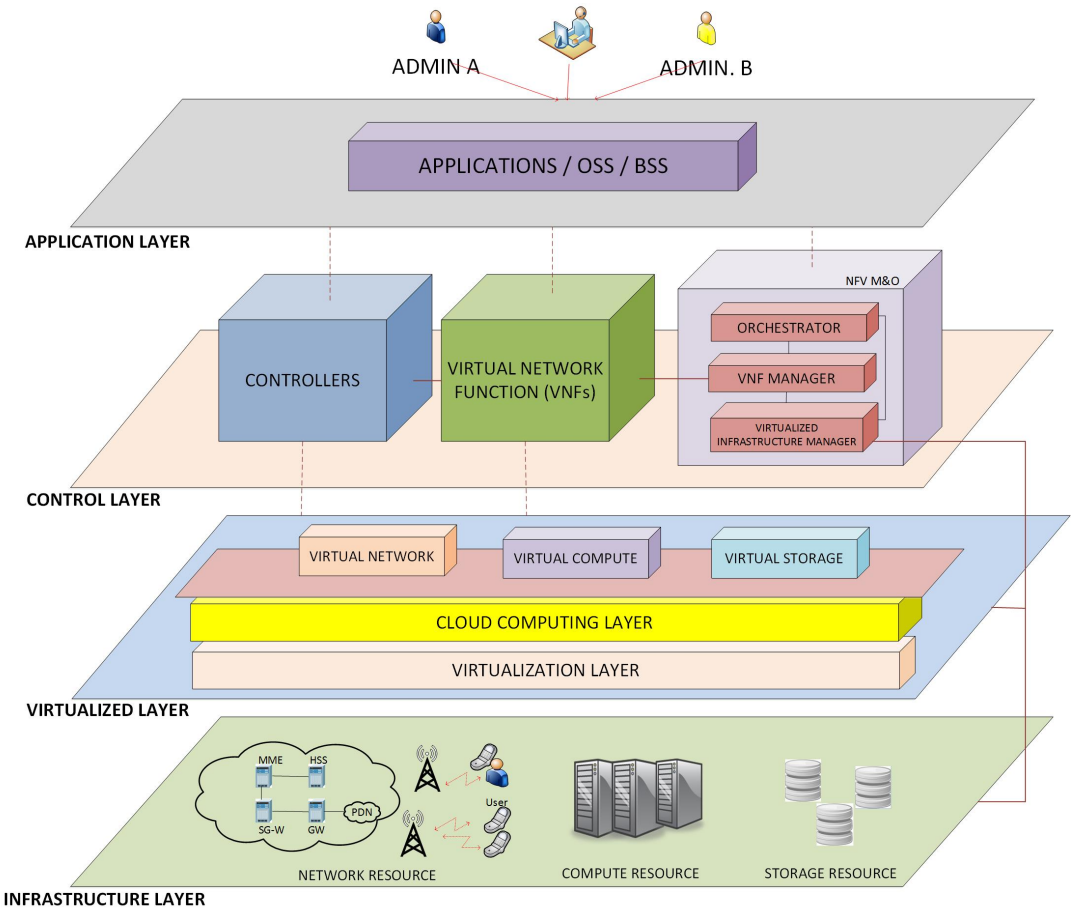

Figure 4.3: Future mobile network architecture

5G will incorporate all of these concepts or part of them. Despite the advantages of this proposal, there are some challenges that need to be overcome in order to successfully combine these technologies. Firstly, the unified definition and standardization in the separation of the data plane and control plane and the provision of virtualized instances will enable the easy development and integration of the future network technologies. In

addition, the complexity of the mobile network elements constitutes a big challenge by itself. At the same time, these kind of systems will require effective pricing schemes and business models with two objectives in mind: i) Customers pay only for the provisioned service and ii) stakeholders receive revenues according to their SLAs. Another important issue is how legacy networks will coexist with new systems, which is still a relatively unexplored field. In the management and orchestration field, significant changes are required not only to improve the processing of data information but also to optimize the deployment and allocation of network resources. A unified management framework could allow enhanced traffic monitoring, provide self-management capabilities and network customization. A virtualized environment faces some issues, such as finding the best place to allocate virtual functions (operator infrastructure or cloud), migration and scheduling process. Mechanisms are also needed to provide load balancing, energy efficiency algorithms, inter-domain capabilities, among others. In parallel, all of these characteristics should be provided in a secure and trusted environment with enhanced capacities to recovery from failures. Moreover, the SDN centralized control or the dynamism of cloud computing are challenges that need to be covered. Table 4.2 shows the challenges and future trends that must be covered in order to fulfill the user needs of 5G networks.

Table 4.2: Current trends and challenges

Requirement	Challenge	Future Trends/Enabler Technologies
System Performance	Provide efficient mechanisms regarding to radio resource provisioning. Improving the capacity of radio resources. Provide super wide bandwidth. Better management of data traffic, interference and mobility levels.	Evolution of RATs. Decreasing the cell size. Millimeter-wave communication Intelligent resource allocation via SDN or SON.
Composite Wireless Infrastructures	The 5G device can choose the most appropriate wireless or mobile technology according their needs (Change between systems).	Enhancement of user devices (Muti-Band-Multi-Mode support). Introduction of intelligent mechanism and SDN control.
Facilitating very dense deployments (Hetnets)	Operators will must provide effective mechanisms to deploy cells of different sizes according to user needs.	Improving the resource capacity through decreasing the cell size. Introduction of intelligent and *Software Defined Radio* (SDR) concepts.
Flexible spectrum management	Improve the spectrum utilization in order to operate in some spectrum bands or channels, while reducing interferences.	Massive MIMO Mechanisms to use unused bands.
Native support D2D Communication	Deploy networks based on interconnected end user devices (machines, sensors,etc). The traffic will be properly assigned without cause congestions.	Introduction of Cognitive Intelligent mechanisms to exchange traffic between users. Smarter end-user devices.
Reduce Capex and Opex	Reduce the average service creation. Dynamic scalability and deployment of services and NFs, while reduce the complexity in planning and configuration tasks.	Resource sharing (Exploring Cloud-RAN, Cloud computing, NFV) Smarter allocation of functional mobile components (SDN, NFV).
Muti-tenancy and multi-service support	Service providers can control the resources deployed in a shared infrastructure (network, computing, mobile resources).	Cloud computing SDN NFV *Mobile Edge Computing* (MEC)
Open Environment	New applications and NFs could be deployed in an open environment, no matter the network hardware and technologies used by operators.	Standardization of SDN and NFV concepts. Introduction of SDR.
Energy efficiency operation	Saving energy per service provided. Nowadays, most of the energy consumption comes from RAN elements.	Introduction of intelligent and SON capabilities taking into account the device status.
Monitoring and Management	Provide self-management and self-optimization capabilities to 5G systems.	Automated management and monitoring functions (SDN, NFV). Takes decisions based on historical record of network status.
Ensuring QoS/QoE and SLA	A 5G user will be able to obtain enhanced services, regardless of the location or network technologies (compared with 4G systems), for several use cases such as emergency situations or network failures.	Enhanced mechanism to monitoring the network status (traffic optimization techniques) via SDN and intelligent mechanism. Automated network configuration to ensure the required need (SDN, NFV).
Charging and billing	Create different user profiles in order customers pay only the required service (pay-as-you-go), while operators bill the respective service.	Introduction of SDN and NFV concepts.

It is important to note the current efforts of initiatives such as 5G Americas [5G 17a], 5G-PPP or NGMN to develop 5G network. They promote not only SDN, NFV and cloud computing adoption but also the study of transversal concepts such as carrier aggregation, massive MIMO, Multi-RAT convergence, spectral and signalling efficiency, among others. It is imperative that telecommunication and network service providers find a consensus to develop solutions, architectures, technologies and standards for the next generation

of infrastructures. The communication paradigm of anytime, anyhow and anywhere will become a reality in the future society.

4.5 Summary

The present chapter reviews the vision of the 5G Generation Mobile Network. For this purpose, the main 5G requirements in terms of User Experience, System Performance, Device and Enhanced Services, New Business models and the Operation and Management are summarized. Then, the 5G KPIs are described. Finally, the future trends and challenges are analysed.

Chapter 5

Related Works

This chapter reviews the research projects and diagnosis capabilities on 5G Networks. This chapter is organized in 4 Sections. Section 5.1 summarizes the 5G research projects on 5G. Then, diagnosis capabilities are described in Section 5.2 Section 5.3 presents the ongoing work related to security and risk management in 5G Networks. Lastly, Section 5.4 summarizes this chapter.

5.1 Research Projects on 5G

5G Networks require customizable, efficient and scalable network infrastructures in order to meet the new user needs and the exponentially-increasing traffic demands, while decreasing the capital and operational expenditures. The SDN concept has been introduced in a broad range of fields, such as QoS, data centers, mobile and optical networks, security, network virtualization, among others [HHB14]. As an instance, Google was one of the first enterprises to incorporate the SDN concept to communicate their internal Datacenter-WAN. Furthermore, there are some projects that allow SDN experimentation by offering scalable testbed infrastructures with research purposes, such as Geant, GENI, Ofelia, Felix, among others [SNC+14].

In particular, the integration of SDN or NFV with mobile networks includes the deployment of virtualized base stations and core components LTE [PWH13], energy efficiency experimentation on WiFi networks, the optimization of very dense and heterogeneous wireless networks [cro17], etc. The next generation of mobile networks could take advantage of the combination of key-enabled technologies to enhance the following areas: (i) the development of radio access (high speed, spectrum efficiency, high mobility, high availability); (ii) improvements in core networks (QoS support, aggregated processes, network slicing, cloud deployment) and (iii) the management and orchestration process (customization of user needs, dynamic allocation of resources, energy efficiency mechanism, manage a big amount of data) [BHL+14] [DGK+13] [AHGZ16].

Different standard organizations leverage the adoption of SDN and NFV concepts in their infrastructures. These organizations have presented the challenges, KPIs and possible use cases in order to cover the above-mentioned areas. As an instance, ONF [ONF17] promotes the adoption of SDN and defines a wide range of use cases, such as inter-cell

interference management, virtual customer edge, network virtualization or data center optimization. Meanwhile, NFV is an initiative of ETSI and telecommunication providers, which proposes the virtualization of the traditional network functions. ETSI-NFV defines nine general use cases [ETS], such as NFVIaaS, VNF-FG, etc. In the scope of mobile networks, NFV promotes the virtualization of Mobile Core Networks and *IP Multimedia Subsystem* (IMS), the virtualization of a mobile base station, the virtualization of the home environment and the virtualization of Content Delivery Networks (CDNs). In the meantime, some open source projects led by the research community have emerged to provide an open environment to test with SDN, NFV and cloud computing, such as OpenNFV (SDN and NFV) [KT217], Floodlight (OpenFlow and OpenStack support) [flo17], OpenDaylight (SDN, NFV and OpenStack) [MVTG14].

With regard to mobile networks, industry manufacturers, telecommunication operators, and related stakeholders are working on the definition of requirements, standardization, regulation and development of future mobile systems, such as 5G-PPP (5G Infrastructure Public Private Partnership) and Next Generation Mobile Network initiative (NGMN). The 5G-PPP [Moh15] proposes solutions, standards and infrastructures to allow the ubiquitous 5G communication. For its part, the NGMN [EHE15] will expect to provide 5G solutions by 2020, within eight general use cases: broadband access in dense areas, broadband access everywhere, high user mobility, massive Internet of Things, extreme real-time communication, lifeline communication, ultra-reliable communication and broadcast-like services. The most outstanding efforts have been made in the 5G research field. A wide range of projects or initiatives will expect to cover the needs of future mobile users. These worldwide initiatives encompass global regions of Asia, Europe and the Americas.

With the aim of promoting the adoption of 5G in Asia, China has launched the IMT-2020 promotion group [IMT17], which manages five working groups: Requirements, Technology, Spectrum, *Intellectual Property Right* (IPR) and Standardization. This is the most important promotion platform related with research and international cooperation purposes. Similarly, coordinated efforts in the 5G area have been launched in South Korea and Japan, the former with the 5G Forum [5G 17b] and the latter with the *Fifth Generation Mobile Communications Promotion Forum* (5GMF) [5GM17]. Both are conducting research projects involving active participants from the government, industry, and academia, in order to facilitate the development of 5G.

Significant efforts have been made in Europe under the support of the European Union Framework Project 7 (FP7) and the Horizon 2020 programmes [Moh15]. On the one hand, FP7 has launched 5G research projects such as METIS, *Mobile Cloud Network* (MCN), CONTENT, T-NOVA, UNIFY, CROWD, etc. On the other hand, Horizon 2020 has financed several research projects (considered by 5G-PPP as Phase 1 Projects) such as 5G-NORMA, METIS II, CHARISMA, SONATA, FLEX5GWARE, SELFNET, among others. In the FP7 context, Mobile and wireless communications Enablers for the Twenty-twenty Information Society Project (METIS) [OBB+14] lays the foundations of 5G networks and promotes the general agreement to design this mobile environment. The first phase of this project (METIS I) includes five big scenarios (amazingly fast, great

Services in a crowd, ubiquitous things communicating, best experience follows you, super real-time and reliable connections) in a use-case driven approach.

The initiative Mobile Cloud Computing (MCN) [mcn17] provides mobile services by means of the combination of three components: mobile network, compute, and storage resources. MCN defines a wide range of use cases, such as RAN on Demand, Mobile Virtual Resources on Demand, Machine Type Communication on Demand, SDN or virtualized EPC, to mention a few. In the same way, the CONTENT project [KT317a] proposes a network infrastructure that enables end-to-end cloud and mobile services. This project provides a virtualized infrastructure based on LTE, WIFI and optical metro networks and introduces the SDN concept in their deployment. CONTENT presents two general use cases: Infrastructure and network sharing (created logical resources) and cloud service provisioning on top of virtual infrastructures (end-to-end).

The integration of SDN with NFV is proposed in T-NOVA [tno17] and UNIFY [uni17] projects. On one hand, T-NOVA provides a framework to deploy NFVs over network infrastructures. The innovation of this project consists of their NFV Apps marketplace, which enables the easy creation, deployment and management of virtual network appliances in a standardized environment. T-NOVA proposes three general scenarios: High-Level Scenario, T-NOVA NFVs, and VNF Chaining. On the other hand, the UNIFY project takes advantage of cloud computing and the virtualization concept to provide a novel network architecture with optimized data traffic flows and the dynamic placement of networking, computer and storage components. This project presents eleven use cases, organized around the following domains: Infrastructure Virtualization, Flexible Service Chaining and Network Service Chain Invocation for Providers.

In the area of SDN and SON, CROWD [cro17] includes these technologies to enhance the coordination process between radio base stations in very dense and heterogeneous wireless networks (Dense Nets). This project allows the network cooperation, the dynamic network configuration, dynamic backhaul reconfiguration, energy optimization, etc. CROWD also presents fifteen use cases divided into two big scenarios: self-optimizing dense networks and Optimized mobility in dense radio access networks.

As part of the Horizon 2020 programme, 5G-NORMA [nor17] is a research project which aimed to provision an adaptive and open 5G infrastructure with capabilities to service customization, enhanced performance and security. To this purpose, this project introduces adaptability capacity to allocate mobile network functions in the most appropriate location and in a short time. Likewise, METIS II [kt317b] presents a novel 5G RAN design, introducing a protocol stack architecture intended to provide a seamless integration of 5G radio technologies. The innovations of METIS II are focused on the spectrum management, air interfaces harmonization, resource management and a common control and user plane framework. The integration of them will support regulatory and standardization bodies.

Other ongoing H2020 projects that combine SDN and NFV technologies are CHARISMA [KT417a] and SONATA [KT417c]. CHARISMA will enable the deployment of an intelligent cloud radio access network (C-RAN) and virtualized *Customer Premise Equipment* (CPE). SONATA will support network function chaining and an enhanced

orchestration process in order to allow service customization.

The provision of innovative hardware and software platforms to support 5G infrastructures is proposed in FLEX5GWARE [KT417b]. This project attempts to develop and prototype key components of 5G networks in the hardware and software domains. The main objective of this project is to deliver a highly reconfigurable hardware platform together with a well suited software platform, over which network elements and devices can be deployed following a modular, efficient and scalable approach. Several components must be deployed as 5G enablers, such as MIMO emulators, high-speed broadband converters, Filter Bank Multi-Carrier (FBMC) transceivers, Low-Density Parity Check (LDPC) codes, etc., with suitable interfaces to allow flexible software-based management schemes.

The integration of SDN, SON, NFV and artificial intelligence is encompassed by the SELFNET [NCC+16] [sel17] project, which introduces intelligent, self-organizing and autonomic capacities to 5G networks, taking advantage not only of SDN and SON but also NFV and Cloud Computing. This project will provide a scalable, extensible and smart architecture to foster innovation and decrease capital and operational expenditures derived from network management tasks. Moreover, SELFNET introduces the SON concept to facilitate the automatic management of network infrastructures. SON solutions are typically classified into three domains: self-protection, self-optimization and self-healing, which are the use cases proposed by SELFNET. Likewise, the COGNET Project [XAY+16] proposes the introduction of machine learning, SDN and NFV in order to enhance monitoring tasks and autonomic network management. COGNET predicts the resource demand requirements and then changes its own configuration based on the network analysis (prediction, frauds, detecting error and security conditions).

Table 5.1 shows the current european use-case driven projects that tackle different 5G requirements, through a combination of SDN, NFV, SON and cloud computing concepts. All of these projects take into account SDN in different domains, such as e-health services, security, service chaining, multimedia optimization, etc.

Last, but not least, it is worth mentioning the research efforts in the Americas, where a group of telecommunication service providers and manufacturers created the 5G Americas [5G 17a], an organization intended to foster the development of LTE wireless technology leveraging the adoption of 5G in the North and South Americas society. At the same time, several activities have been conducted by academia. For instance, the *Berkeley Wireless Research Center* (BWRC) involves university, industry, government and other research stakeholders focused on exploring innovations in wireless communication systems based on radio frequency and millimeter wave technologies, which are its main challenge to develop reconfigurable radio architectures. Likewise, the *Broadband Wireless Access and Applications Center* (BWAC) involves around fifty research centers with the aim to collaborate with the industry in the creation of innovative and scalable wireless networks.

Table 5.1: Research projects in mobile networks

Project Name	Related Technologies	Main Objective	Scenarios/Use Cases
MCN [mcn17]	SDN, Cloud Computing	Enhanced traffic processing by means of the separation between radio hardware and packet forwardinghardware.	Cloud Computing for Mobile Network Operations, End-To-End Mobile Cloud
T-NOVA [tno17]	SDN, NFV	Design and implementation of an integrated architecture for the automated provision and management of VNFinfrastructures.	High-Level Scenario, NFVs, Service chaining
UNIFY [uni17]	SDN, NFV	The development of an automated and dynamic service provision platform, based on a service chaining architecture	Infrastructure Virtualization, Flexible Service Chaining, Network Service Chain Invocation for Providers
CROWD [cro17]	SDN, SON	The creation of technologies to support dynamic network functionality configuration and fine, on-demand, capacity tuning.	General scenario
5G-NORMA [nor17]	SDN, NFV	The development of an adaptive, customizable, secure and efficient mobile network architecture to deal with complex traffic demandfluctuations.	Multi service, Multi tenancy
CHARISMA [KT417a]	SDN, NFV	The creation of an intelligent and hierarchical routing and paravirtualized architecture to enhance end-to-end services.	General scenario
SELFNET [sel17]	SDN, NFV, SON, Cloud	The design and implementation of an autonomic network management framework to achieve self-organizing capabilities in managing 5G network infrastructures, leveraging an improvement in the overall userexperience.	Self-healing, Self-protection, Self-optimization
COGNET [XAY+16]	SDN, NFV, Machine Learning	Dynamic adaptation of the network resources (virtual network functions), while minimizing performance degradations and fulfill SLA requirements.	Situational context, Just in time Services, User Centric services, Optimized Services, SLA Enforcement, Collaborative Resource Manage

5.2 Diagnosis Capabilities in 5G Networks

5G network envisages an architecture able to cover three main domains [EHE15] i) enhancement of radio capabilities to enable the spectrum optimization, the interference coordination and cost-effective dense deployments, ii) provisioning of an effective network management environment to create and deploy a common core to support several use cases in a cost-effective manner, and iii) simplification of the system operations by means of automated procedures, where the introduction of new capabilities or network functions should not imply increased complexity on operations and management tasks. In order to tackle these requirements 5G networks take advantage of the separation between data and control plane (network programmability) offered by SDN architectures, the deployment of virtualized network functions, the scalability and flexibility in the service provisioning

based on cloud environments, enabling high capacity and massive communications (cognitive radio, carrier aggregation, Machine to Machine Communication), spectrum and resource optimization (millimeter wave and massive MIMO) and intelligent capabilities provided by artificial intelligence or self-organization concepts [ARS16], [LWC+16].

In particular, the introduction of analysis and intelligent capabilities [BGMB14], [5G 17a] could be applied to several domains such as autonomic network maintenance, automation in the provisioning of services, prediction and remediation of congestion or queue utilization, detection of security threats, improving network efficiency, multi cell coordination, provisioning of high QoS and QoE for services, etc. For this purpose, analysis and intelligent capabilities allow to response to network problems based on pattern recognition, the dynamic smart selection of the best location where the services can be deployed or migrated, sharing and releasing of resources based on forecasting methods, building of context awareness models based on real time information from the network, its devices and applications. In order to provide intelligence and facilitate the decision-making process, some tasks must be performed. On one hand, analysis stage is intended to perform the identification of network situations and events. These situations do not necessarily imply (a priori) a harmful nature. On the other hand, the decision-making task determines if a specific situation is a risk for the network health, or its components, and then it performs the respective countermeasures.

In this context, traditional approaches apply different analysis and reasoning techniques, such as Bayesian Networks (BN) [FN01], in order to provide intelligent to common network management tasks. However, these models are not sufficient to guarantee the network performance according to SLAs and future 5G user needs [EHE15]. There are some proposals to address the data analysis in 5G systems and its elements such as access and radio components [DGK+13], [MMZ+16], network devices [TPL+16], cloud elements [KT317a] and resource allocation [KS16]. In [AHNR16], a prototype to perform mobile network analysis based on *Markov Logic Network* (MLN) and semantic web technologies is presented. This approach allows the optimization and network status characterization but does not explain how it covers heterogeneous data sources. For its part, Imran et al. [IZAD14] proposes a framework to provide a full view of network status based on machine learning and big data concepts. To this end, their proposal predicts the user behaviour and dynamically associates the network response to the network parameters. However it is does not specify how to deal with SDN or NFV components.

Meanwhile, there are reports [EHE15], [BMMR+15] and projects [EN1], [NCC+16], [nor17], [KT417a], [XAY+16], that introduces analysis capacities to cover 5G requirements. In this way, METIS Project [TPL+16] takes into account SON concept in order to provide a new level of adaptability to 5G infrastructures. Meanwhile, 5G-NORMA [nor17] introduces adaptive capacities to allocate network functions based on user and traffic demands over time and location. CHARISMA project [KT417a] deploys an intelligent cloud radio access network and end devices. For its part, 5G-Ensure [EN1] proposes a 5G secure system based on risk assessment and mitigation methodologies. COGNET [XAY+16] takes into account Machine learning, SDN and NFV to provide dynamic adaptation of network resources. For its part, a whole approach to address not only

analysis component but also the whole cycle of incident management in 5G networks is proposed in [BLVCMV+17]. This work applies the three stages of processing information of Endsley Model [End88] to 5G Networks: perception, comprehension and projection. In the perception phase the monitoring and collection of different metrics from network infrastructure (and its elements) are performed. Then in comprehension stage, the association and correlation of this information are performed in order to provide enhanced metrics to be analysed (projection phase). The analysis component includes the diagnosis and prediction of the whole state of the system. In general terms, these proposals aid to tackle 5G Requirements but they do not offer a generalized approach able to take into account several kind of metrics from heterogeneous data sources that is the case of SELFNET Project [sel17].

5.3 Security and Risk Management in 5G Networks

The new 5G design principles are intended to support an exponential increase of connected devices and, consequently, the data traffic moving through the network. In contrast to traditional mobile architectures, 5G requests a clear separation between data and control planes, a global vision of the network and a dynamic/customizable control of the mobile network operations. For this purpose, innovative technologies, such as SDN and NFV, have been extended to wireless and mobile platforms. In this way, the operators are not limited by the use of CLI for individual and remote access. Instead, the administrator can create software or "network applications" to dynamically control the network behavior. However, the use of autonomous incident management systems that take advantage of these new paradigms is limited. In this context, the main challenge is the coordination between the virtual monitoring elements, allocated in different nodes in the infrastructure, and the response or mitigation procedures through the execution of actions in virtual functions. Similarly, mobile SDN/NFV-enabled architectures are limited by the lack of integrated schemes capable of analyzing large volumes of data, detecting potential risks and diagnosing their causes. Furthermore, the management systems should enable the definition, organization and handling of the different risks, assets and priorities without compromise the security and quality of service.

To the best of our knowledge, there are few studies that survey Security and Risk Management in 5G Networks. In [MMZ+16], a context aware framework for the next generation of MCN is proposed. This work introduces a "Context Generation and Handling Function" to provide enriched processing information from radio and core elements, taking into account two key-enabled 5G technologies (SDN and NFV concepts). Meanwhile, a recently published threat report has been conducted by the European Union Agency for Network and Information Security (ENISA) [BMMR+15]. This work reviews the potential security in SDN/5G networks, considering not only SDN but also NFV and Radio fields. This report identifies the network assets and the security threats, their related challenges and risks. It also describes the existing security methods and provides good practices for 5G systems. These works could be considered part of the initial research of 5G Security Management. However, these are limited in scope. On one

hand, the ENISA Report [BMMR+15] identifies only the assets and threats to SDN/5G environments (no architecture proposal has been done). On the other hand, Marquezan et al. [MMZ+16] propose a single network function that monitors radio and access elements but does not take into account other 5G components such as virtualization or application layers. Meanwhile, the 5G-Ensure Project is intended to cover security requirements in 5G Networks. The proposed architecture will provide a trustworthy 5G system, offering reliable security services to customers by means of the development of a set of non-intrusive security enablers such as privacy, security network management, and trust, among others [SA117]. As part of its proposal, 5G-Ensure defines a Risk Assessment and Mitigation methodology in order to evaluate security concerns in 5G systems, based on NIST-SP-800-30 and ISO 27005 standards [EN1]. Although 5G-Ensure covers a wide range of security issues on 5G Networks, this project is still at an early stage and does not keep in mind the concept of actuators, which have been introduced in our proposal in order to mitigate possible risks and deploy corrective measures.

5.4 Summary

This chapter presents the advances on the development of 5G technologies and its current research projects. In particular, the Diagnosis Capabilities and its applicability fields on this kind of environments. Furthermore, the research in security and Risk Management applied to 5G Networks is presented. This chapter reviews the main advances of self-management networks based on SDN/NFV. Similarly, the SELFNET and its components are described.

Chapter 6

SELFNET SDN/NFV Self-Organized Networks

This chapter reviews the main advances of SDN/NFV-based self-management networks. Similarly, the Self-Organized Network Management in Virtualized and Software Defined Networks Project SELFNET is described. In the same way, each component of the framework is described.

This chapter is organized in 10 Sections. Section 6.1 introduces this chapter. Section 6.2 reviews the network management with SDNNFV. Section 6.3 presents the self-organized network management framework for SDN/NFV. The next sections gives details of the layers and sublayers of SELFNET. Section 6.4 presents the infrastructure layer. Section 6.5 discusses the data network layer and the Section 6.6 presents the SON Control layer. Section 6.7 reviews the the SON autonomic layer. The Section 6.8 studies the NFV orchestration and management layer. The Section 6.9 presents the SON Access layer. Lastly, section 6.10 summarizes this chapter.

6.1 Introduction

The management and customization of network services have been limited by the rigidity of traditional network architectures and the increasing of both capital and operational expenditures. Actually, the resolution of common traditional network problems, such as link failures, security attack, QoS or QoE degradation, bottlenecks, among others, requires the direct involvement of network operators. The manual re-configuration of the equipments or even the installation of new equipment (router, *Network Address Translations* (NATs), firewalls) compromises the normal operation of the network and causes the disruption of the SLAs. Similarly, the creation of innovative value-added services is limited by the closed and proprietary hardware/software, and in some cases, all infrastructure may belong to the same provider.

Those limitations makes traditional network architectures unfeasible to meet the requirements of todays users, enterprises, and carriers. The solution proposed to solve these challenges is following the advances reached by computing, where developers can create their own applications using a high level programming language. The programs can

be executed in several equipments thanks to the abstractions of resources provided by the Operating Systems. In this context, the SDN and NFV appears as a promising strategy to reach those objectives. SDN proposes the decoupling of data and control planes in network devices enabling their independent development and evolution and a centralized view of the network. NFV promotes the migration from typical network equipments (DPI, firewall, load balancers) to a software package or NF that can be instantiated in a virtualized infrastructure. Both architectures are complementary and potentially could be integrated to provide an open network environment for developers.

Furthermore, the exponential growth of mobile devices and content together with the advent of cloud services bring additional challenges to operators and service providers. A radical decrease of integrated network management operations without negatively affecting the QoS/QoE and security is required. Similarly, a new model that integrates the access and management of mobile resources is promoted. The future 5G networks are expected to expose not only typical mobile broadband but also a heterogeneous, simplified and unified control. The network management expenditures can be reduced through automation of operations. In this context, a scalable management framework that includes data mining, pattern recognition, learning algorithms to reduce operation expenditures is challenging. This chapter also describes the status and recent advances on this research area.

The SELFNET [sel17] uses SDN/NFV principles to provide smart autonomic management of network functions in order to resolve network problems or improve the QoS. SELFNET integrates the self-management paradigm with the use of data mining, learning algorithms, pattern recognition to identify the network behaviour and including 5G mobile architectures. For this purpose, the SELFNET architecture is composed of well defined layers: Infrastructure, Virtualized Network, SON Control, SON Autonomic and Access Layer. Within SON Autonomic Layer, the Monitor and Analyzer sublayer is one of the most challenging operations. Monitor and Analyzer, on its turn, is divided in three modules: Monitoring and Discovery, Aggregation and Correlation, and Analyzer.

6.2 Network Management with SDN/NFV

The SDN and NFV principles offer several advantages over traditional network management architectures. Several consortiums formed by operators, academia, service providers have focused their effort on development of novel management architectures over virtualized environments. In Table 6.1, relevant SDN/NFV-based management projects are described.

6.3 SELFNET Self-Organized Network Management for SDN/NFV

The SELFNET H2020 project will design and implement an autonomic network management framework to provide SON capabilities in new 5G mobile network infrastructures. By automatically detecting and mitigating a range of common network problems, currently manually addressed by network administrators, SELFNET will

Table 6.1: Network management projects based on SDN/NFV

Project	Domain	Description	Application Scenario
CROWD [cro17]	SDN, SON	The project aims to bringing density proportional capacity in heterogeneous wireless access networks. Similarly, it focuses on guaranteeing mobile users QoE, optimizing MAC mechanisms and proportional energy consumption. In this way, it enhance the traffic management in dense wireless networks	Traffic Management
5G-NORMA [nor17]	SDN, NFV	The project focuses on providing adaptability of a resource in efficient way. The framework handle fluctuations in traffic demand resulting from heterogeneous and dynamically changing service portfolio. The novel network functions offer resource-efficient support of varying scenarios and help to increase energy-efficiency	Multi-service scenario, Multi-tenancy scenario
MCN [mcn17]	SDN	The project focuses on the enhancement of traffic traffic processing by means the separation between radio hardware and packet forwarding hardware	SDN environment
UNIFY [uni17]	SDN, NFV	The project aims to develop an automated, dynamic service creation platform through the creation of a service model and service reaction language. It will enable the dynamic and automatic placement of networking, computing and storage components across the infrastructure. Similarly, the orchestrator will include optimization algorithms to ensure optimal placement of elementary service components across the infrastructure	Infrastructure Virtualization, Flexible Service Chaining, Network Service Chain Invocation for Providers
T-NOVA [tno17]	SDN, NFV	The project focuses on the deployment on *Network Functions-as-a-Service* (NFaaS) over virtualised Network/IT infrastructures. For this purpose, it design and implement a managementorchestration platform for the automated provision, configuration, monitoring and optimization of virtualized resources. Moreover, the SDN is also used for efficient management of the network infrastructure	High-Level Scenario, VNF Chaining Scenario

provide a framework that can significantly reduce operational costs and consequently improve the user experience [sel17] [NCC+16].

By exploring the integration of novel technologies such as SDN, NFV, SON, Cloud computing, Artificial Intelligence, QoS/QoE and next generation of networking concepts. SELFNET will provide a scalable, extensible and smart network management system. The framework will assist network operators to perform key management tasks such as automatic deployment of SDN/NFV applications that provide automated network monitoring and autonomic network maintenance delivered by defining high-level tactical measures and enabling autonomic corrective and preventive actions to mitigate existing or potential network problems. SELFNET will address three major network management concerns by providing self-protection capabilities against distributed network attacks, self-healing capabilities against network failures, and self-optimization features to improve dynamically the performance of the network and the QoE of the users. The facilities provided by SELFNET will provide the foundations for delivering some of the 5G requirements defined by 5G-PPP consortium.

In this context, Figure 6.1 illustrates the architecture of the SELFNET framework. The architecture is based on five differentiated layers with the following logical scopes: Infrastructure Layer, Data Network Layer, SON Control Layer, SON Autonomic Layer,

NFV Orchestration & Management Layer, SON Access Layer. In the following sections, each layer will be described.

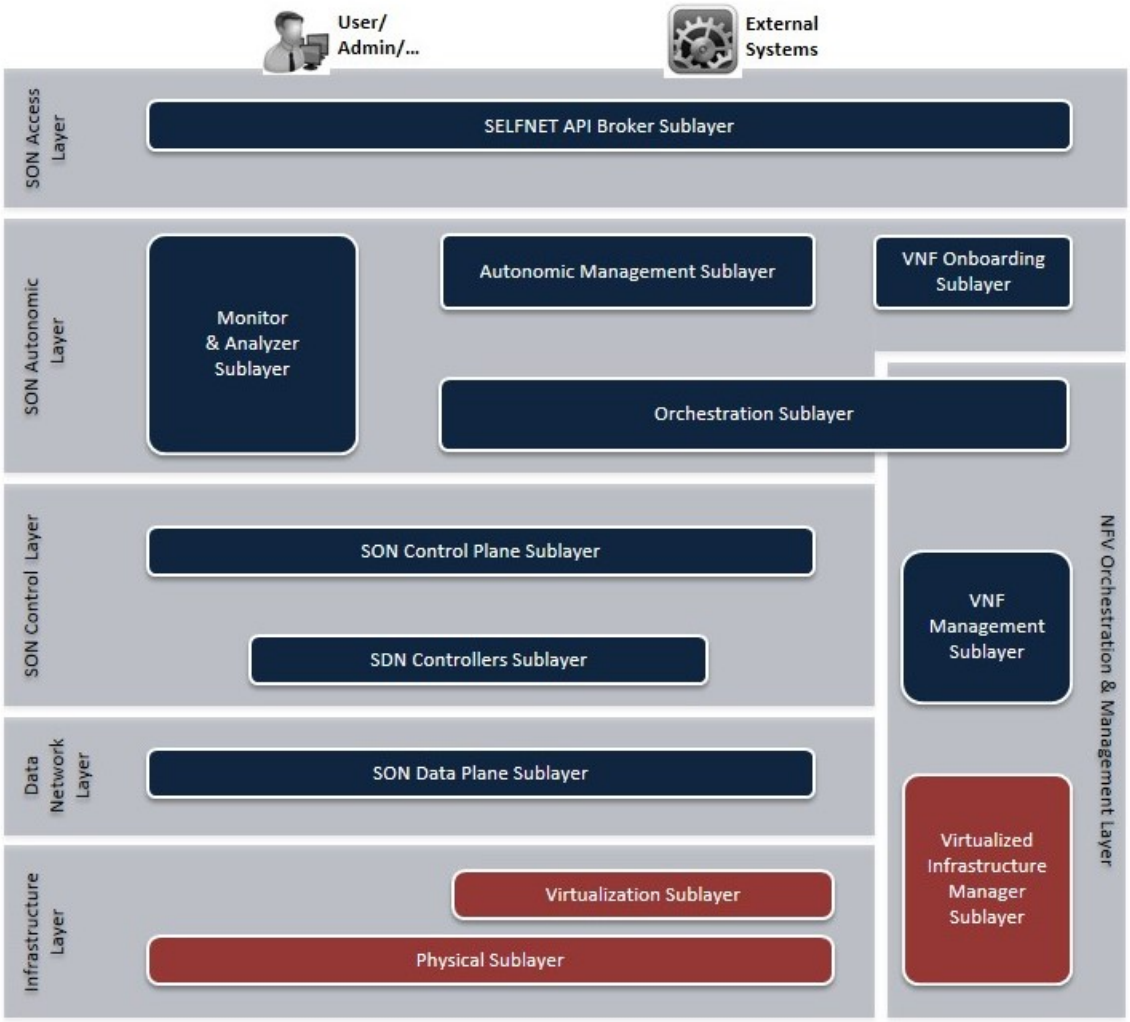

Figure 6.1: SELFNET architecture overview [NCC+16]

6.4 Infrastructure Layer

This layer provides the resources required for the instantiation of virtual functions (Compute, Network and Storage) and supports the mechanisms for that instantiation. It represents the NFVI as defined by the ETSI NFV terminology [ETS13a]. In order to achieve its functionality, two sublayers will be defined: Physical Sublayer and Virtualization Sublayer.

6.4.1 The Physical Sublayer

The Physical Sublayer includes the physical resources required to provide computation, networking and storage capabilities over bare metal. Since SELFNET is designed to include 5G networks, the physical elements follows a mobile edge architecture in which

operators can deploy the operational and management services. The MEC proposed by ETSI [PNC+14] is depicted in Figure 6.2. It proposes the edge nodes geographically separated from the data centre. In this way, the services could be deployed close to the user if operational or management services require high performance. Moreover, the integration of edge deployments (e.g. C-RAN) within the MEC discontinue the typical rigidity and allows the customization of services. Similarly, it is considered that the connectivity between the elements enables virtualization capabilities following the advances on 5G.

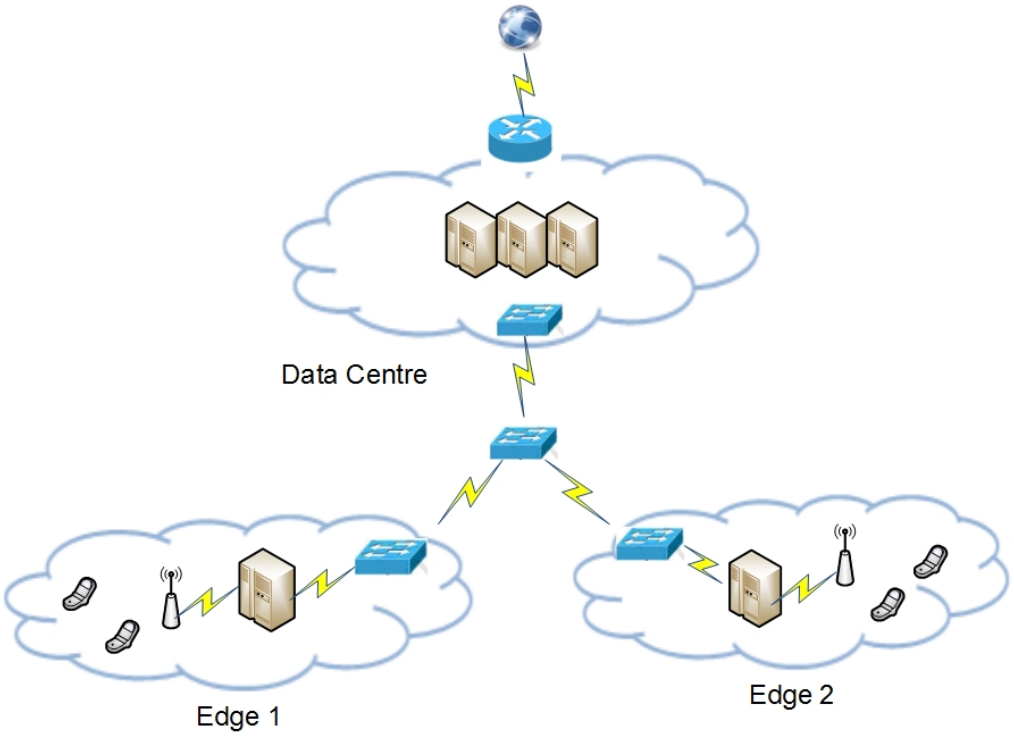

Figure 6.2: Physical layer of MEC for 5G architectures

6.4.2 The Virtualization Sublayer

The virtualization sublayer enables the sharing of the available resources between different users or services. It offers several advantages, such as the isolation, reliability, adaptability and control of resources. However, the main disadvantage can include the performance penalti of the virtualization related tasks. In this context, recent advances on virtualization technologies have reached the expectations on 5G infrastructures on the performance of virtualized workload with intensive Input/Output (I/O) [AGHE+15]. In other words, the performance penalti of using virtualization can be considered as negligible for modern devices. In SELFNET, the virtualization sublayer includes the use of virtual switches used to connect virtual Machines allocated on the physical resources.

6.5 Data Network Layer

In this layer, the different functionalities of the networks functions are located and interconnected in a designed topology. The NF includes the instances required for the normal operation of the virtual infrastructure and those created by SELFNET as part of the SON functionalities. Because of the edge and centre location are fully virtualized, the NFs can be dynamically allocated in both locations.

The Data Network Layer also provides multi-tenancy support. Multitenancy enables the sharing of resources among different tenants, each with their own administrative domain and business requirements. In 5G architectures, the resources can be acquired by a telco Alliance and the resources are shared according to their needs. In this context, a specific telco administrator is not able to administer resources of other telco or intercept traffic provided by other traffic.

6.6 SON Control Layer

This layer includes the elements responsible of collecting data from different virtualized sources (SON Sensors) and the functions that execute actions into the network (SON Actuators). The SON Sensors and SON Actuators are controlled by the SON Autonomic Layer, which provides network intelligence. Similarly, the SON Control Layer deal with the control plane in SDN architectures. In other words, it translates an autonomic network-wide policies into specific network elements configurations.

6.6.1 SDN Controller Sublayer

The SDN Controller Sublayer implements a logically centralized controller (e.g. SDN control plane). It provides the governance of the network elements and controls the network functions. The SDN controller uses the information about the actual network behaviour and then can enforce the rules configured in the network elements. In this way, the traffic passing through such network elements can be dynamically modified. The interface between the controller and the network elements depends on the network device capabilities. For instance, protocols such as OpenFlow, OVS, OFConfig, *Network Configuration Protocol* (NETCONF) can be supported by an SDN Controller. The different SDN controller services are deployed by a set of SDN controller Applications or SDN-Apps. Examples of SDN-Apps includes routing protocol, forwarding protocol, filtering protocol or tagging protocol. In northbound, the SDN Controller provides an API to enable the remote management, configuration and monitoring of the controller behaviour.

6.6.2 SON Control Plane Sublayer

The SON Control Plane instantiate the different network functions NF running in the virtualized infrastructure. In SELFNET architecture, which aims to provide self-organized capabilities, there are two types of NFs: SON Sensors and SON Actuators.

- **SON Sensors**. It collects data related to network activities. The collected information includes metrics related to global traffic (e.g. link status, bandwidth) or specific metrics (e.g. DPI, QoS on a video streaming related to a specific data flow). The operators and service provider can develop different sensors according to their needs.

- **SON Actuators**. It executes a specific set of actions on the traffic circulating in the network. The actions depends on the application developed by the service providers. For instance, if the systems detects a DDoS attack, a SON actuator can automatically block the specific attack source. For its part, if the system detects a QoS degradation, another SON Actuator can optimize the network flow increasing the priority or bandwidth.

6.7 SON Autonomic Layer

This layer in responsible to provide the network intelligence. The information collected from sensors is used to diagnose the network situation. Then, the actions to accomplish the systems goals are determined and executed. The main components of SON Autonomic Layer are described as follows.

6.7.1 Monitor and Analyzer

This sublayer collects the information provided by sensors. Then, this information is aggregated and correlated in order to extract the relevant information. Then, the analyser use the relevant information to detect network situations (botnet detected, QoS/QoE degradation, DDoS attack, link failure). The whole process is organized in thee steps: Monitoring and Discovery, Aggregation/Correlation and Analyzer.

- **Monitoring and Discovery**. It collects the data sent by the SON Sensors. For this purpose, when a new Sensor is instantiated, it receives the notification and instantiation details and establishes a connection in order to receive the corresponding metrics. Moreover, it also receives the information provided by the physical and virtual sublayers. Then, the information is stored in a database in order to be processed by upper layers.

- **Aggregation and Correlation**. It performs the correlation and aggregation of the information stored in the monitoring and discovery database. This process involves additional actions, such as the data normalization, verification and removement of redundant information. At the end of this stage, only relevant information will be processed by the Analysis module.

- **Analyzer**. Its main purpose is the comprehensive analysis of the relevant information provided by Aggregation and Correlation. The analysis also includes the prediction of future network problems. The network problems are known as *Health of Network* (HoN) due the global vision or system-level scope of the analysis. For

this purpose, it takes advantage of several prediction, pattern recognition algorithms and big data techniques. The trending and predicted values for the metrics enable the application of proactive and reactive actions in the system. At the end of this stage, the network events are sent to the autonomic manager in order to establish the corresponding actions in the network.

6.7.2 VNF Onboarding

It acts as a repository of the different NFs. In this sublayer, the available network functions are stored and their capabilities are disseminated to the other sublayers. Similarly, the service providers can design, create and update their own applications. In this conext, the encapsulation of NFs follows the recommendations of the ETSI MANO framework for the NFV [ETS14]. Consequently, the NFV Manager (VNFM) is the key component for the lifecycle of SON sensors and actuators. The VNFM lifecycle exposes a common set of primitives for the automated instantiation, configuration, re-configuration and termination of the different VNFs. A common API enables service providers the easy design and development of their solutions. Once a solution (NFs) is onboarded, the autonomic manager can use their capabilities to provide the new service (sensor/actuator).

6.7.3 Autonomic Manager

It uses different algorithms to diagnose the root cause of a network problem in terms of the HoN metrics provided by the Analyzer. Once the cause is detected, the autonomic manager uses the available NFs provided by the VNF onboard to decide the best reaction strategy or a countermeasure (e.g. deploy a new balancer, firewall or DPI). Then, the taken actions are sent to the NFV orchestration and Management Layer. The related tasks are organized in three well defined modules.

- **Diagnoser**. It diagnoses the root cause of the network situations notified by the analyzer. For this purpose, it uses the information available on Monitor & Analyzer sublayer (topology, sensor data, HoN metrics) and takes advantage of stochastic algorithms, artificial intelligence, data mining to estimate the location of the source of the problem. Then, the root cause is notified to the Decision Maker.

- **Decision Maker**. It takes the incoming diagnosis information and decides a set of reactive and proactive actions to be taken into the network in order to avoid the detected and emerging network problems, respectively. Similarly, it also takes advantage of the integration of artificial intelligence algorithms to determine the responses or tactics to be taken. The taken decisions are notified to the action enforcer.

- **Action Enforcer**. It provides a consistent and coherent scheduled set of actions to be taken in the infrastructure. In other words, it validates, organizes and refines the tactics to avoid conflicts, duplications and nonsense order of actions. At the end of this stage, a high level description of the location, type of SELFNET SON Actuators, related configuration parameters are transferred to the orchestrator.

6.8 NFV Orchestration & Management Layer

This layer is responsible of the control and chaining of the different NFs in the virtualized infrastructure. The architecture follows the ETSI MANO [ETS14] recommendations and, consequently, it is composed of: Orchestration, VNF Management and VIM. As described in the Section 6.7.2. The VNF Management operations are partially developed in the VNF Onboarding. The other operations are described as follows:

- **NFV Management & Orchestration**. It is responsible of receiving the set of actions of the Autonomic Manager and orchestrate the NFs in the available virtual resources. The coordination and schedule of the enforcement of different actions is executed by the interaction with the virtual infrastructure manager.
- **Virtual Infrastructure Manager VIM**. It is responsible of organize and provide the virtual resources for the instantiation of the different NFs. The VIM interacts with the physical and virtual infrastructure to ensure the availability of resources and perform the automatic deployment of services.

6.9 SON Access Layer

This layer will provide an appealing and intuitive interface that provides different monitor and operation capabilities depending on the authorized users. In this way, the users can check on the current health status of the SELFNET operations. Similarly, the Access API will list the SON Sensor and Actuator currently deployed in SELFNET as well as the logging of an messages in order to enable a wider view of the SELFNET status. This interface is used by external actors such as BSS or OSS.

As described in the previous sections, SELFNET aims to be an independent and autonomous solution that acts mitigating or solving network problems without any actions from real users. In this way, the SON Access Layer also provides users with the study of the actions taken by SELFNET allowing the validation of corrective measures of the applications.

6.10 Summary

This chapter summarizes the self-management advances with SDN/NFV principles. Then, the Self-Organized Network Management in Virtualized and Software Defined Networks Project SELFNET is proposed. Next, the different SELFNET layers and sublayers are described. The infrastructure layer, data network layer, SON control layer, SON autonomic layer, NFV orchestrator & management layer and SON access layers are discussed.

Part II

Description of the Research

Esta parte del documento corresponde a la
aportación original y exclusiva de la Tesis Doctoral.
This part of the document corresponds to the
original and exclusive contribution of the Doctoral Thesis.

Chapter 7

5G Situational Awareness Framework

This chapter proposes a novel architecture for incident management on 5G. The approach combines the conventional risk management schemes with the Endsley Situational Awareness model. It also considers the dynamicity of 5G environments, the countermeasure tracking and the role of context when decision-making is done. The proposal takes into account all layers for information processing in 5G mobile networks, ranging from infrastructure to the actuators responsible for deploying corrective measures.

The rest of this chapter is outlined as follows: Section 7.1 introduces this chapter. Section 7.2 describes the situational awareness and information security concepts. Section 7.3 gives details on the architecture for the incident management on 5G networks. In section 7.4 describes the main characteristics regarding to the analysis and decision-making stage. Finally, Section 7.5 summarizes this chapter.

7.1 Introduction

Nowadays, information security management plays a major role towards achieving the objectives and goals of companies and organizations. Traditionally, it has been carried out by implementing guidelines, standards and platforms that aim to protect their resources and assets ISO/IEC 27000 [fStIEC05], NIST-SP 800 [oST07], CVSS-SIG-First [oIRT16], MAGERIT [dHyAP12], ITIL and COBIT [POK13], etc.). However, these proposals have shown shortcomings when they are implemented in dynamic scenarios, where the context plays a very important role in decision-making [WAMS14]. This is the case of network-based monitoring environments, and more specifically, those that implement 5G technology, where the assets and events are considered highly dependent on the environment. As a solution to this problem, some authors have adopted incident management methodologies capable of handling information in a much more cognitive way, and, therefore, facilitating their understanding through contextual analysis. Worthy of special mention are those based on constructing Situational Awareness (SA) of the protected environment by applying the Endsley's model, where the perception, comprehension and projection of the system status are kept in mind [End88]. The

adaptation of this paradigm to the management of information security in networks has led to the coining of the term Network *Network Security Situational Awareness* (NSSA) [LM15]. Despite, however, its effectiveness having been proven in existing networks, it has not yet been considered to meet the challenges posed by 5G technologies. This chapter introduces the key ideas and concepts for incident management on 5G environments, combining Endsley Situational Awareness Model and traditional risk management guidelines.

7.2 Situational Awareness and Information Security

As defined by Endsley, the term SA refers to "the perception of the elements in the environment within a volume of time and space, comprehension of their meaning and the projection of their status in the near future" [ESHC98]. Usually, this definition is simplified as "knowing what's going on so you can figure what to do" [Ada93]. Thus, it is clear that its aim is to facilitate decision-making based on what is happening and its projection [End88]. In order to acquire Situational Awareness, Endsley proposes three stages of information processing: perception, comprehension and projection; the first conducts the tasks of monitoring and identification of incidences, the second their analysis and association, and the last predicts the evolution of the state of the system. As shown in Figure 7.1, once relevant situations are detected, the countermeasures to be applied are decided and executed. It is important to highlight that there is feedback between action/decision levels and the Situational Awareness; in this way, the countermeasures and their impact on the system are tracked. The observed results have implications on future decisions and facilitate the use of advanced diagnostic methods [LM15].

Incident management based on Situational Awareness has been implemented in very different areas, among them smart grids [DAKM15], power generation [NNL15] or vehicular collision avoidance systems [MPTSF16]. In [SV15], a method for defining the critical information and the relevant information quality elements that are required to build the *Shared Situational Awareness* (SSA) in disaster response is suggested. The adaptation of the Endsley has proved particularly effective in complex and dynamic environments [DRAP15], where the diagnosis is highly dependent on the context in which incidents are reported, reaching to play an essential role in the fight against cybercrime. Many of these contributions are collected in [BBS14], where the predominance of issues related to risk management in emergency situations, industrial systems and networks is observed. As discussed in [WAMS14], they improve the three most repeated deficiencies of the Information Security Risk Management: 1) Information security risk identification is commonly perfunctory; 2) Information security risks are commonly estimated with little reference to the current situation; and 3) Information security risk assessment is commonly performed on an intermittent, non-historical basis (a conventional security risk assessment scheme can only give a "snapshot" of the risks of the information systems at a particular time [SM10]). In order to bring solutions to these problems, but without losing focus on the *Information Security Risk Management* (ISRM)/ISRM basis, several publications approach the combination of both paradigms. This is the case of

[WAMS14], [NLZ14] where the Situational Awareness is acquired, taking into account the definition of risks, assets and their impact posed by the different standards and platforms for ISRM implementation [fStIEC05], [oST07], [oIRT16], [dHyAP12], [POK13].

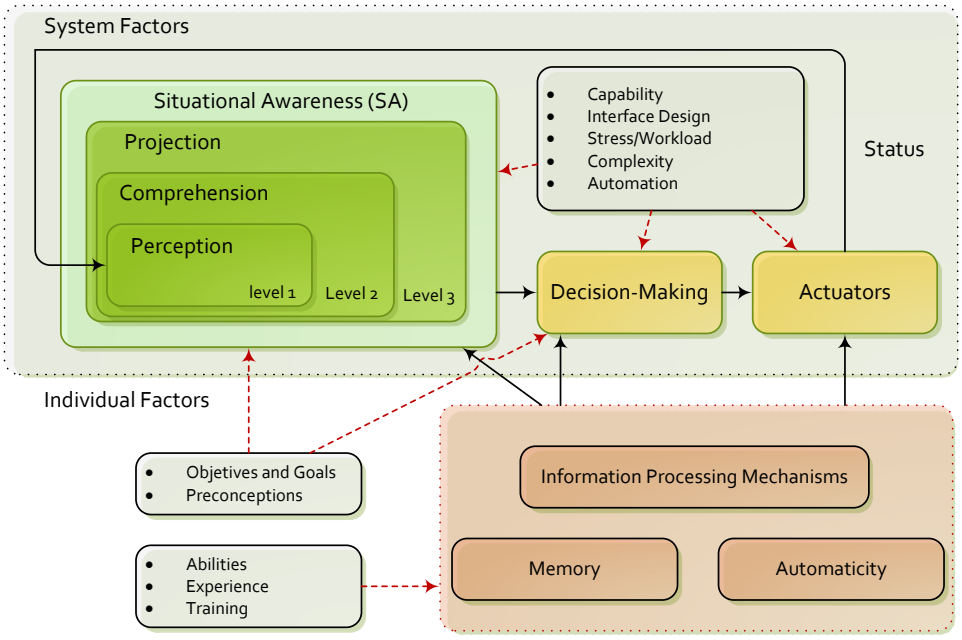

Figure 7.1: Endsley model for situational awareness

7.3 Information Security Architecture for 5G

The proposed architecture is mainly focused on autonomous risk management of 5G mobile infrastructures based on SDN/NFV architectures. In order to establish the coverage and limits of this approach, the following addresses the assumptions and requirements of the following design:

- The elements responsible for monitoring and executing mitigation actions (e.g., virtual functions) are compatible with the SDN/NFV paradigm. If the elements follow the traditional architectures, a compatibility layer is assumed. This additional layer can use the available configuration options to emulate a SDN/NFV enabled element.
- The communication between the different modules of the framework must be performed through secure channels.
- The information provided by the monitoring elements (low level metrics, alerts) are considered reliable.
- The Risk Analysis and the corresponding Situation Awareness procedures are strongly isolated from the data plane forwarding. In other words, the resources (network, storage and computing) used for the operation of the framework belong

to administrative domain and, consequently, do not modify the capabilities of the 5G forwarding elements.

- The functional modules are extensible and can be implemented using distributed architectures in function of the available management resources and the size of the managed infrastructure.

The proposed architecture presents the synergy between 5G risk analysis and the Endsley model, and is depicted in Figure 7.2. The model describes four functional layers: Virtual Infrastructure and Sensors, Monitoring and Correlation, Analysis, and Decision-Making and Actuators. The Virtual Infrastructure executes the data forwarding engine and Sensors monitors the different metrics of the network. For its part, the perception, comprehension and projection principles of the Situational Awareness are applied in the Monitoring, Correlation and Analysis modules. The Decision-Making and Actuators complete the circle with the execution of proactive and reactive actions to optimize and solve problems located in the virtual infrastructure. The following sections of this chapter focus on the description of the Virtual Infrastructure/Sensors and Monitoring/Correlation.

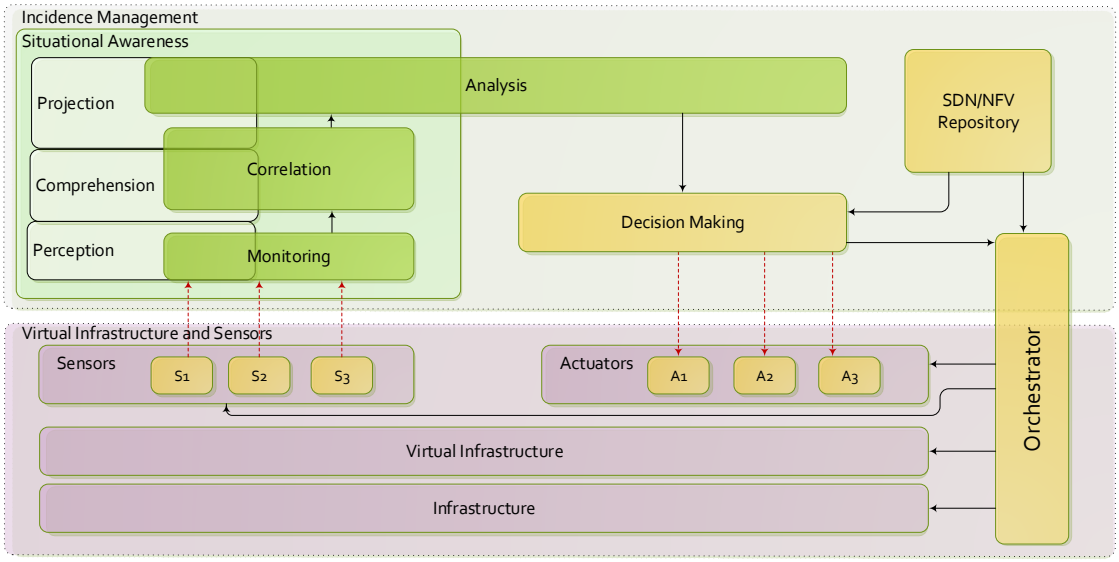

Figure 7.2: Architecture for risk management in 5G

7.3.1 Virtual Infrastructure and Sensors

The main purpose of this layer is the abstraction of the different hardware/software elements running in the mobile infrastructure and enabling the monitoring of low-level metrics related to the network behavior/status. Its developments includes the innovative designs principles of 5G networks: decoupling of data and control planes, virtualization of mobile functions, and a complete integration with cloud computing environments. In this way, the SDN architecture promotes the separation between data and control planes

in mobile infrastructure (base stations, links, servers, gateways, DPI, among others). The network administrator is not limited by the traditional private/closed hardware/software system and, consequently, the data forwarding engine can be customized. Similarly, the virtualization layer enables the dynamic allocation of virtual resources based on the user requirements. For its part, the NFV approach proposes the implementation of the different services (e.g., firewall, DPI, QoS optimizer, load balancer) as virtual software functions that can be instantiated in different points of the virtualized infrastructure. As a result, the architecture considers two software applications types: SDN-Apps and NFV-Apps. SDN-Apps is executed in software programs to control the data plane in network devices and the NFV-Apps are virtual functions that can be instantiated in virtualized elements to develop a particular service. In the proposed architecture, the sensors are specialized NFV-Apps capable of monitoring different metrics on the system. Traffic analyzers, QoS analyzers, and anomalies/botnet/DDoS attack detectors are examples of sensors. These sensors (NFV-Apps) can be instantiated in different locations of the virtual infrastructure and reconfigured depending on the requirements of upper layers. Consequently, the system is able to increase the surveillance in suspected hazardous areas and establish quarantine regions.

7.3.2 Monitoring and Correlation

The monitoring module collects the information provided by lower layers (Virtualized Infrastructure and Sensors) and applies aggregation/correlation techniques to simplify the further analysis tasks:

- *Monitoring (Data Collection).* The main objectives are the gathering and management of the information from all data sources, and facilitating their access to upper layers. Monitoring tasks would be able to actively poll different sources to collect real-time statistics, providing highly accurate and low overhead traffic measurement [TSK+17]. This module also controls the registration and access process of new sensors. The collected information is organized in efficient data structures, taking into account the large amount of data to be processed. In this regard, two scenarios were considered. In the first scenario, the sensor sends a report to the monitor when it detects relevant information (alerts, link failures, memory or CPU overload). In the second scenario, the monitor requests information (whenever necessary) to the sensors in order to facilitate the aggregation and analysis tasks (virtual topology, available links, among others).

- *Correlation.* It is responsible for the first abstraction level of information processing, in which, in order to have a global view of the network status, correlation and aggregation processes are executed. Information considered as redundant or non-sensitive is discarded. As an example, in case of multiple alerts received from each device belonging to the same affected area, a single alert is displayed with the affected topology. Due to the dynamism offered by virtual environments, in contrast to the rigidity of the physical elements, network topology is expressed as

an extended or increased graph ($G_a(V_a, E_a)$), which models virtual nodes (V_a) and links (E_a) located in the physical infrastructure [SKW+15], [CRB09]. Likewise, as a result of correlation and aggregation operations, the received low-level metrics can be expressed or translated into high-level metrics, also known as Health of Network (HoN). For example, transmission data rate (Mbps), delay (ms) and jitter (ms) of data in streaming video, collected by the sensors at different points in the network, can be expressed as an overall perception of quality of service QoS/QoE, quantified by the measurement of the *Mean Opinion Score* (MOS).

7.4 Analysis and Decision-Making

This section describes the principal characteristics of the components related to analyzing the gathered information, decision of countermeasures and their deployment.

7.4.1 Analysis

The analysis component performs identification of network situations from metrics provided by the aggregation module and reaches diagnoses that contribute in decision-making tasks. In general terms, the analysis studies any aspect related with the incidences reported by the 5G use cases and the risks that could compromise the system requirements. In this context, situations are divided into two main groups: events and risks. Events are defined as situations that occur within 5G mobile networks which a priori do not display harmful features but are useful in diagnosis. The events are grouped into four categories: discovery, removal, modification and notification, which are described as follows:

- *Discovery events.* It includes all situations related to incorporation of new assets to the system. For example, this occurs when incorporating new nodes into the network, establishment of new connections between previously existing resources, or deployment of new virtualization layers. Each time a discovery event is communicated, the asset inventory is updated.

- *Removal events.* Unlike discovery events, removal events indicate situations related to the elimination of 5G resources. These are the cases of deletion of assets, removal of connections between nodes, or elimination of virtualization resources. As in discovery events, each time a removal event is communicated, the asset inventory is upgraded.

- *Modification events.* They include every situation related with the modification (not removal) of an existing resource. For example, this occurs when varying the location of the asset (i.e., changes to IP address, MAC address, etc.), and changes between communication protocols or software updates. As in the previous cases, modification events involve changes to the asset inventory.

- *Notification events.* They report specific situations in the network that are not related to changes in the assets inventory, such as periodic reviews of the bandwidth status, presence of unused resources or requests for special configurations.

On the other hand, risks are inherently damaging, and they may be inferred from network mapping or directly reported by the use case agents. An example of the first case is the identification of bottlenecks, congested regions or resources depletion. In the other case, a striking example relates to defensive use cases, where security NFV-Apps (IDS, honeypots, etc.) directly reports intrusions such as malware spreading or denial of service threats. The bases of 5G situation analysis are shown in Figure 7.3. They include: detection, risk assessment, asset inventory management, risk map, prediction, diagnosis and countermeasure tracking. The following briefly describes the most important features of each of them.

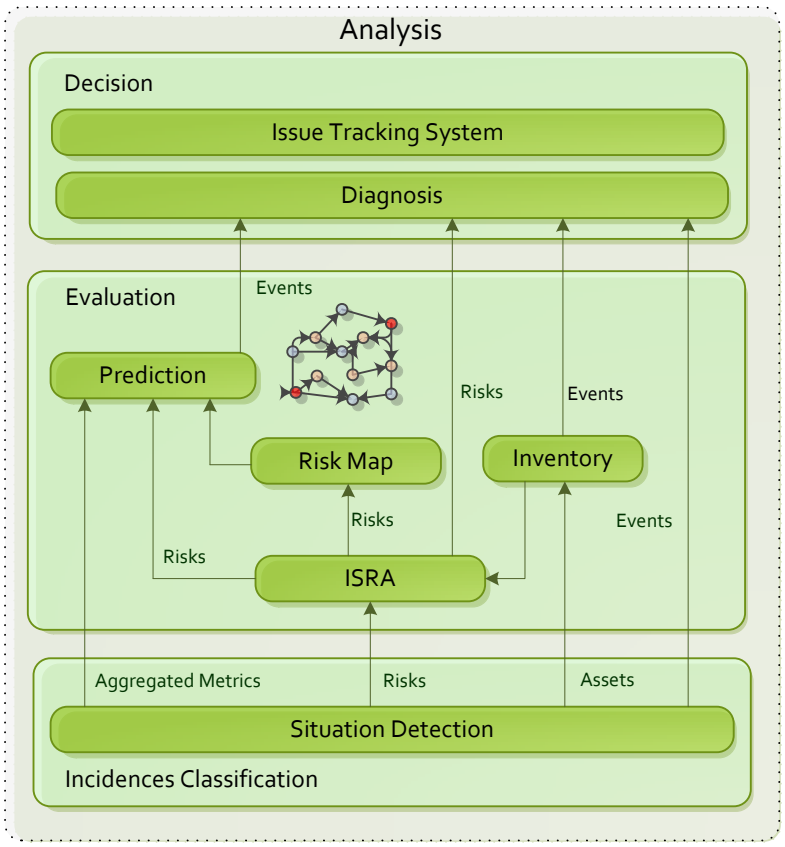

Figure 7.3: Situation analysis on 5G

7.4.1.1 Detection

The detection module is the connection between monitoring/aggregation tasks and the functions for understanding the information. Its inputs are aggregated high-level metrics built from correlated data and reports of situations directly issued by the sensors. After processing this information, the detection module builds the primitive situations (events or risks) to be analyzed. The rules to infer situations from the perceived data are provided

by security operators and determined after the risks/events identification. Note that the same combination of metrics could trigger different situations; because of this, the proper management of the detection component implies adaptation of expert systems, especially those based on rules.

7.4.1.2 Risk Assessment

Risk assessment combines some of the most widely spread strategies approved by the research community for ISRM, among them the guidelines ISO/IEC 27000 [fStIEC05], NIST-SP 800 [oST07], CVSS-SIG-First [oIRT16] or MAGERIT [dHyAP12]. The approach to identify risk factors assumes the basis on ISO/IEC 2700 series and NIST-SP800. However, since these are mainly aimed at general purpose risk assessment, they lack specificity; in particular, they do not take into account the 5G design principles, infrastructure or requirements. Because of this, they must be adapted to the 5G mobile networks circumstances. Another obstacle is that they are based on metrics that are too simple. In order to improve the ability to understand the impact and facilitate decision-making, as well as consider a more current model, a group of advanced measures similar to those proposed in CVSS-SIG-First should be adapted. Thereby, a larger amount of characteristics on the potential vulnerabilities should be studied, thus assuming the union of three metric sets that contain intrinsic (base), temporal and specific (environmental) features. On the other hand, approaches like MAGERIT provide alternative ways to calculate risks, which may be particularly useful in certain use cases. In general terms, the risk assessment component may be integrated as part of the detection module or could be deployed completely independently. Bearing in mind the taxonomy of ISRM approaches [SSABC16], it is recommended that its development considers qualitative assessment criteria, service-based perspectives, vertical valuation and propagated measurement.

7.4.1.3 Asset Inventory

The asset inventory builds and manages a list of resources or assets to be considered at the risk valuation step. Due to the ability of the 5G mobile networks to automate the deployment of new services and network devices depending on their status, this component plays a critical role in the analysis of the gathered information. The new assets are detected at the monitoring layer and are reported by discovery events. When the existing events are updated, the monitor layer emits modification or removal events. For the proper functioning of the proposed architecture, it is very important to ensure coherence between the list of assets and the real network resources.

7.4.1.4 Risk Map

This component builds and manages a risk map of the network incidences considering aggregated high-level metrics, events and the inferred risks. The risk map is mainly considered in the following situations: prediction of threats spreading, the establishment of quarantine regions, identification of the best spots to deploy mitigation actions, and

recognition of the source of the attacks. As in the case of the asset inventory, there must be coherence between the list of assets and the current network resources. The risk map is built considering the network map, so its proper development implies upgrades in real-time of the network connectivity and its status (throughput, congestion, transmission delays, availability, etc.)

7.4.1.5 Prediction

Prediction facilitates the anticipation of future complications, such as congestion of certain network regions, inclusion of large amounts of new assets or spreading of cyberattacks. The general scheme for forecasting SA studies variations on their contents by modeling the sequence of observations as time series, as it is shown in Figure 7.4. Each observation includes information about every SA feature, such as the network map, risk levels or recent incidents monitored over a period of time. Because of this, the prediction module is not only capable of anticipating particular situations, but also the whole future SA, thus providing general and particular overviews about everything that occurs in the system. Note that the observations are delimited by a fixed period of time. A common alternative to this fixed value is to consider more specific features, such as workload or number of situations reported. Both cases imply advantages and disadvantages, but the delimitation by time periods poses a more intuitive method. In order to avoid overloading of storage systems, after a reasonable period, the oldest observations are discarded to make way for the new ones. In this way, a sliding window of size N that gathers the observations is applied. This boundedness is important for ensuring that the implemented algorithms are computable, avoiding the case $N \to \infty$ (i.e., the worst case), which leads to holding an infinite amount of information.

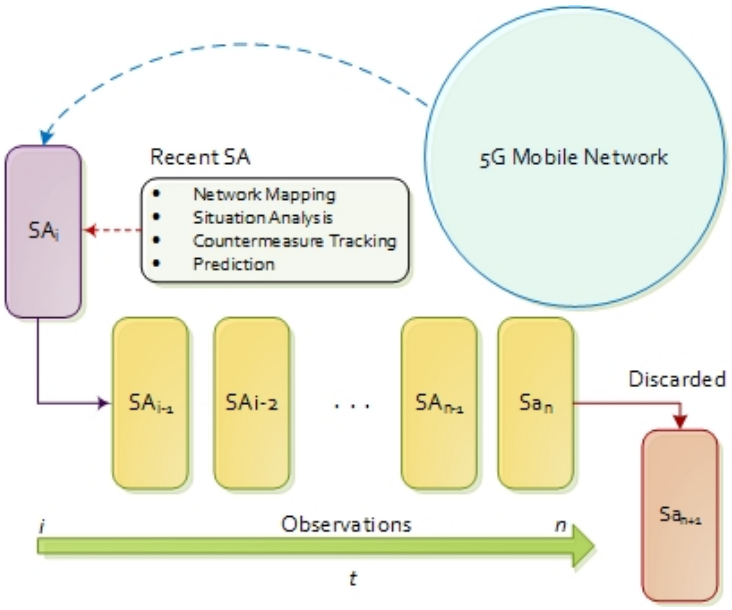

Figure 7.4: Situational awareness prediction

7.4.1.6 Diagnosis

Diagnosis performs advanced analysis of risks and their assessment, impact, projection and network status. This allows for identifying complex situations that can be difficult to detect from the lower levels of data processing. For example, the diagnosis component should be able to recognize botnets by analyzing the relationships between the risk in network devices compromised by malware and the discovery of surrounding traffic anomalies. Given that the proposed architecture assumes a service-driven vertical risk analysis model that considers propagation, it is important to determine two diagnosis criteria: particular threat level and propagated threat level. The particular threat level related with a risk is its severity. This value is useful to manage isolated situations, but it does not take into account what happens in the surrounding area. On the other hand, the propagated threat level considers all the risks detected in a region and their relationships. There are several publications that address the calculation of propagated risks, where the *Bayesian Network* (BN) are the most widespread solutions [ABR16], [SSH15]. An example of this methodology is shown in Figure 7.5, where several particular threat levels related to each other are normalized and pooled, thus allowing for inferring the propagated risk in the lower situations.

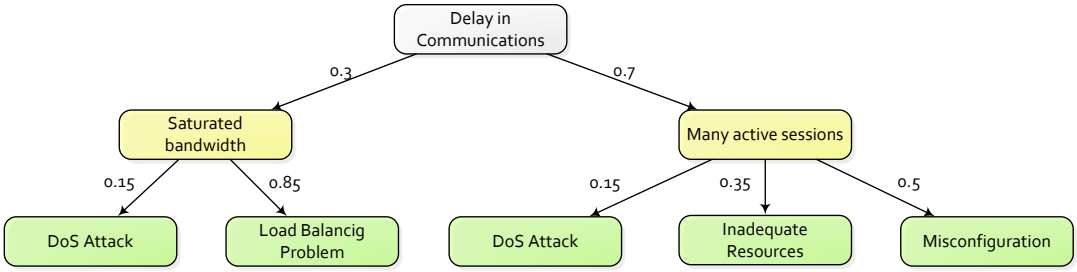

Figure 7.5: Example of bayesian network in network diagnosis

7.4.1.7 Countermeasure Tracking

The countermeasure tracking component conducts comprehensive monitoring of the actions proposed by the decision-making modules for dealing with situations. This allows for identifying ineffective countermeasures that may lead to new diagnosis or prevention of counterproductive situations. On the other hand, the countermeasure tracking stage allows for the development of an immune memory. Therein, all situations that have been resolved or are being processed are stored. This allows us to know how a problem was previously solved, offering added value to the decision-making. In addition, it provides information about all similar problems that are being processed, which may facilitate the correlation of incidences. In order to enhance the countermeasure tracking tasks, the proposed architecture implements an *Issue Tracking System* (ITS). The ITS assigns a ticket to every situation tracked in the protected environment. Tickets are running reports on a particular problem, which contains their nature, history and other relevant data. They are continuously analyzed to provide real-time status of the situation, and they record all the countermeasures implemented over time, as well as their effectiveness.

The ITS is illustrated in Figure 7.6, where the original situation to be treated is provided by a use case, or it is detected from the network map. Then, the incident is analyzed, and a token with the results is sent to the decision stages. If countermeasures should not be applied, the ticket is close; in this case, it is assumed that the problem is fixed, and the solution is stored in the immune memory. Otherwise, countermeasures are applied. Then, the evolution and effectiveness of such actions are studied and added to the ticket history field; as in the previous case, the progress is also stored in the memory. The resulting ticket is sent back to the decision-making module, and this process is repeated until the problem is fixed.

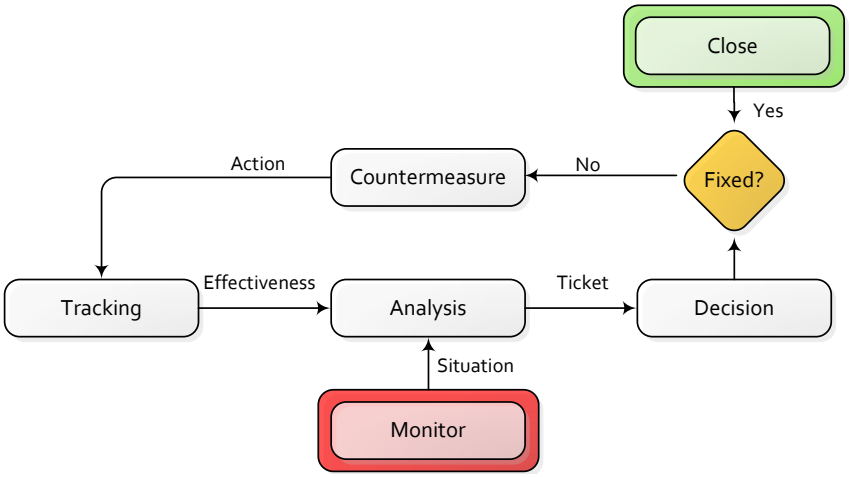

Figure 7.6: issue tracking algorithm

7.4.2 Decision-Making and Actuators

The decision-making component addresses the problem of mitigating the network situations that may disrupt the normal operation of the network elements and the services provided. In particular cases, it can take also part in the performance optimization process of the services offered by the mobile network, thereby playing an active role in tasks such as load balancing or management of traffic with multimedia content. To this end, the decision-making processes receive information from the analysis stage (mainly from diagnosis and countermeasures tracking) and then select a set of responses to be executed. According to ISO/IEC 7498-2 [Int89], such correctives are referred to as security safeguards, and they are involved in prevention, mitigation and source identification. The strategy for selecting optimal countermeasures must balance the cost of implementation of the safeguard and the achieved reduction of the incidence impact [Int89]. The available actions are distributed as NFV-Apps, so that these constitute a large repository of potential countermeasures. The network components that execute the safeguards are known as actuators. An example of their implementation is the mitigation of distributed denial of service attacks: when the analysis component performs a diagnosis related to this category of incidences, the report includes information about the compromised assets, impact, prediction and the attack vector.

In the example, the first step of decision-making is to prevent the spreading of the threat by deploying firewall NFV-Apps (specific actuators), thus taking into account the predictions of their distribution. The second step is to mitigate the threat by implementing honeypot NFV-Apps and the malicious traffic redirection towards sinkholes. Once the impact of the attack is minimized, the last step is identifying the sources by applying IP traceback algorithms [AR14], from which they could be blocked or alerted. The deployment of the various actions is coordinated by an orchestrator agent, which ensures that the virtual resources that implement the countermeasures are available and do not affect the system performance.

7.5 Summary

This chapter introduces a novel architecture for incident management on 5G Mobile networks, which combines the foundations of the traditional risk management guidelines with the Situational Awareness model published by Endsley. It covers all layers of information processing in 5G networks, from the infrastructure to the actuators responsible for implementing mitigation actions. The basis for the identification, monitoring, analysis, decision-making, prediction and countermeasure tracking are also introduced.

Chapter 8

SELFNET Analyzer Module

This chapter presents the design of SELFNET Analyzer Module, with the main objective being to identify suspicious or unexpected situations based on metrics provided by different network components and sensors. The SELFNET Analyzer Module provides a modular architecture driven by use cases where analytic functions can be easily extended. It includes the description of diagnosis and prediction capabilities in 5G environments and how it is being applied in current research and projects. The rest of this chapter is outlined as follows: Section 8.1 describes the main characteristics and analytic capabilities of SELFNET Project and its relation with the situational awareness model. Section 8.2 outlines the design principles, requirements and the architecture of the Analyzer Framework as a whole. Section 8.3 shows this framework as a black box, emphasizing its data inputs and outputs. Section 8.4 defines the data specification of the use case descriptors. Section 8.5 illustrates examples of the data specification and their workflows. Section 8.6 describes the orchestration of the Analyzer Framework. Finally, Section 8.7 summarizes this chapter.

8.1 Analyzer Module vs. Situational Awareness Model

SELFNET H2020 Project [sel17] aims to provide an autonomic network management framework for 5G mobile network infrastructures through the integration of novel technologies such as SDN, NFV, SON, cloud computing and artificial intelligence. SELFNET enables both autonomic corrective and preventive actions to mitigate existing or potential network problems while providing scalability, extensibility and reduce capital expenditure (capex) and operational expenditure (opex). These capabilities are provided through a layered architecture and a use-case driven approach, as is detailed in [NCC+16]. SELFNET architecture addresses major network management problems including self-protection capabilities against distributed cyber-attacks, self-healing capabilities against network failures, and self-optimization to dynamically improve the performance of the network and the QoE of the users.

For this purpose, SELFNET defines two kind of advanced network functions: i) sensors to monitor specific information from the network and ii) actuators to address or mitigate possible problems. In particular, the network intelligence is provided by SON

Autonomic Layer. This layer collects metrics related to the network behaviour and uses that information to infer the network status. Then, it decides the actions to be executed to accomplish the system goals. The SON Autonomic Layer is composed by two sublayers: i) Monitor and Analyzer Sublayer and ii) Autonomic Management Sublayer. The Monitor and Analyzer Sublayer follows the Endsley Situational Awareness Principles. Monitoring and Discovery, Aggregation and Correlation and Analyzer modules corresponds with the Perception, Comprehension and Projection functions as is shown in Figure 8.1.

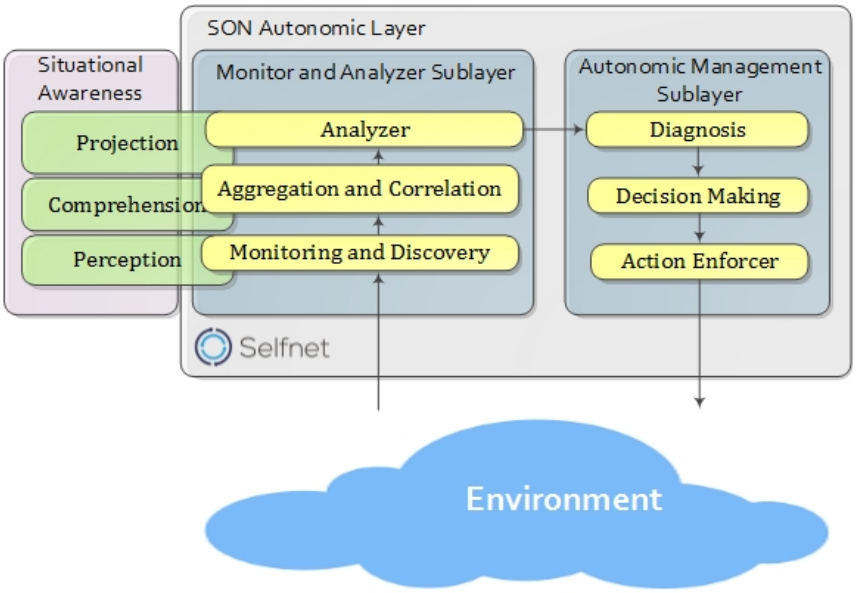

Figure 8.1: Endsley vs. SELFNET autonomic layer

Regarding the Analyzer Module, its main goal is to infer data from the monitored metrics in order to facilitate proactive responses over the network infrastructure (i.e. enhance diagnosis and decision-making tasks). Therefore the Analyzer Module is the first step to provide intelligence to the system, where complex conclusions should be reached by reasoning about knowledge provided by the Monitoring/Aggregation stages and the definition of each use case. Because of this, the Analyzer Module distinguishes three great information processing activities: Pattern Recognition, Reasoning and Prediction. The achieved conclusions are described in the form of symptoms related with each use case. Bearing this in mind, it is possible to assert that the Analyzer Module provides a symptom-oriented Situational Awareness bounded by the situations defined for each use case.

8.2 SELFNET Analyzer Module Design

In this section the design of SELFNET Analyzer Module is detailed. It also describes the initial assumptions, the requirements, the design principles as well as the Analyzer architecture.

8.2.1 Initial Assumption and Requirements

The following describes the most relevant requirements and the main initial assumptions considered in the design of the Analyzer Module:

- *Scalability.* The approach must be allowed to add new capabilities (extensibility), according to SELFNET design principles. For this reason, the integration of additional analytic functionalities are done via plugins.
- *Use case driven.* Given the heavy reliance of the tasks performed with the characteristics of use cases, the basic definition of the observations to be studied (Knowledge-based objects, rules, prediction metrics, etc.) are provided by the use case operator, thus being the Analyzer Module scalable to alternative contexts.
- *Knowledge acquisition.* It is well known that the most common disadvantage of the expert systems is the initial knowledge acquisition problem. Hence, to have skilled operators in novel use cases with the ability to properly specify rules is not always straightforward. This document does not address the issue of the innate knowledge acquisition. Our approach assumes that the use case knowledge-bases are provided by skilled operators or by accurate machine learning algorithms.
- *User-friendly symptom definition rules.* The definition of proper rule-sets is a tricky business. Thus, even the skilled operators often do coherence/ambiguity mistakes. In order to mitigate these problems, the configuration and definition of new use cases should be user-friendly, as well as the scheme for building new rule-sets.
- *Uncertainly.* Classical logic permits only exact reasoning. It assumes that perfect knowledge always exists, but this remains far from the SELFNET reality. In order to improve the quality of the conclusions, the Analyzer Module manages the knowledge bearing in mind uncertainty. This is particularly appropriate for certain analytic features, such as studying observations based on decision thresholds or confidence intervals. In addition, closing the door on possible stochastic dependent definitions is against the SELFNET design principles, as these could be the keys to properly specify future use cases.
- *Filtering.* Initially, the filtering of symptom reports is not considered. Because of this, every inferred symptom, regardless of nature or uncertainty, is transmitted to the diagnosis/decision-making stages, where their impact and relevance are properly assessed.

8.2.2 Design Principles

The following design principles and limitations lay the foundation of the Analyzer framework, as well as the implementation of its internal components:

- *Big Data.* In order to deal with huge and homogeneous datasets, Big Data provides predictive algorithms, user behaviour analytics, and aggregation/correlation

functionalities [SKIW17]. These capabilities are mainly taken into account in monitoring and aggregation tasks. The Analyzer Module deals with aggregated and correlated metrics, hence reducing the amount of information to be analyzed. In our approach, the implementation of Big Data technologies to handle all this information is optional, leaving the decision of integrate these tools at the mercy of the SELFNET administrators, which driven by a better awareness of the use cases and the monitoring environment are more able to decide whether they are counterproductive or beneficial [HHG16]. Because of this, our contribution is compatible with both Big Data and conventional techniques.

- *Stationary monitoring environment.* According to Holte et al. [Hol93], in a stationary monitoring environment, the characteristics and distribution of the normal observations to be analysed match the reference sample population considered in the Analyzer learning processes. If the monitoring environment distribution is able to change representatively, it is considered non-stationary. Another problem that may reduce the quality of the analytics is the presence of gradual changes over time in the statistical characteristics of the class to which an observation belongs. In the literature this fluctuation is known as concept-drift. These problems are discussed at length in [DRAP15]. The assumption that the Analyzer Module operates on a stationary environment brings a simple and efficient solution, but prone to slight failures when the changes occur. On the other hand, to consider a non-stationary monitoring environment improves accuracy, but entails new challenges, among them: detection of changes, implementation of model/regression updating techniques, identifying when the calibration must be completed or selection of the samples that will be taken into account in new trainings. Given the complexity that this implies, the Analyzer Module assumes a stationary monitoring environment. The non-stationary approach will be part of future work.

- *High dimensional data.* The analysis of high dimensional data bears in mind data whose dimension is larger than dimensions considered in classical multivariate analysis. As indicated by Bouveyron et al. [BBS14], when conventional methods deal with high dimensional data they are susceptible to suffer the well-known curse of dimensionality, where considering a large number of irrelevant, redundant and noisy attributes leads to important prediction errors. Hence operate with this data implies the need for more specific and complex algorithms. In terms of SELFNET this means that the vector of HoN is large enough to consider the implementation of specific methods adapted to optimize the processing tasks of this kind of information. A priori there are no signs of SELFNET requiring processing an important amount of High Dimensional Data. Therefore, this proposal does not take into account differences between conventional and High Dimensional data, assuming that the aggregation tasks will be able to optimize the amount of attributes to be analyzed.

- *Supervision.* SELFNET training mode. The analytical methods based on modeling/regression assume that new knowledge can be inferred from observations, by a prior learning stage. The learning process often requires reference data which

allows identifying the most characteristic features of the monitoring environment, such as rules, boundaries, incidence matrices, direction vectors or basic statistics. Given the complexity involved in designing a SELFNET training mode, this approach describes how the information needed for the construction of new models is obtained.

- *Centralized design.* To assume a centralized approach lead us to pose a general purpose scheme where the onboarding of new use cases is completely configurable by specification, and which does not requires updating the implementation (see Figure. 8.2). Therefore the centralized approach is not dependent on the characteristics of the use case, so it is highly scalable and allows performing tasks efficiently (avoiding redundancy). However, its design and the description of use cases is complex. On the other hand, the distributed approach includes an additional component for each use case in which specific pattern recognition and prediction methods are implemented via plugin. The preprocessing, selection and symptom discovery mechanisms have general purpose. In essence, this second approach is easy of design, but completely use case dependent; each time a new use case is onboarded, the Analyzer implementation must be updated. Due to the large impact on the scalability that this entails, the centralized approach is considered hereinafter.

- *Data encapsulation.* The greatest challenge in designing the SELFNET Analyzer approach is the requirement of dealing with unknown data. It is possible to assume that use cases do not provide clear enough information about the characteristics of the information to be analyzed (in fact, several future use cases are completely unknown). At the specification stage, use cases operators tend to provide good qualitative information about the metrics to consider, but may overlook details about their quantitative nature: data type, domain, range, restrictions, etc., which is what in the first instance, will be considered in the analysis tasks. Furthermore, quantitative information is very use case dependent. In order to subtract relevance to quantitative details (which are the backbone of the aggregation/correlation tasks), and thus facilitate the incorporation of new use cases by definition of general purpose descriptors, the SELFNET Analyzer is driven by data encapsulated in two levels of abstraction: quantitative and qualitative parameters (see Figure. 8.3). The first one is independent of the use cases, and allows designing a centralized analysis framework valid for any type of data specification. On the other hand the qualitative parameters gather information directly related with SELFNET and the use case to which they belong (metric name, source, location, tenant, etc.). This data is mainly required for aggregation/correlation, diagnosis and decision making.

8.2.3 Analyzer Module Architecture

In Figure. 8.4 the architecture of the Analyzer Module is illustrated. It is centralized, and distinguishes the following eight core components: (1) Pattern Recognition, (2) Prediction, (3) Adaptive Thresholding, (4) Knowledge-base, (5) Inference Engine, (6) Memory, (7) User Interface and (8) Uncertainty Estimation. The set (4),(5),(6),(8) is related with

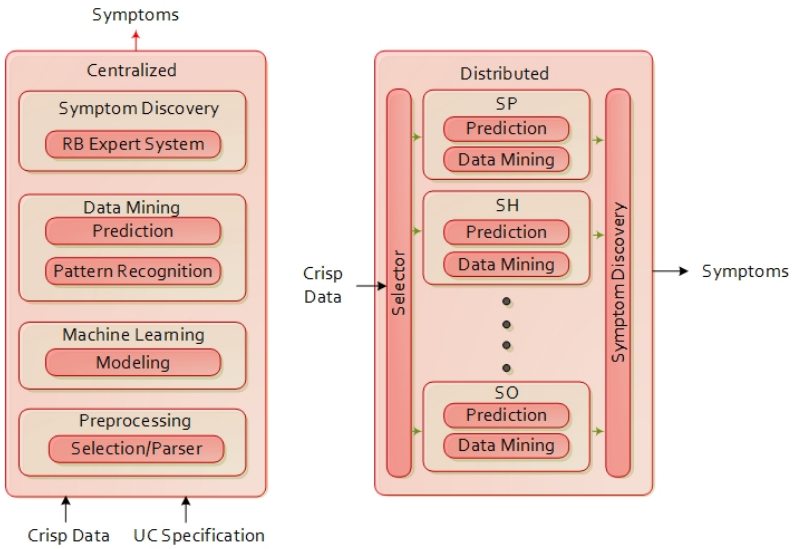

Figure 8.2: Centralized and distributed architectures

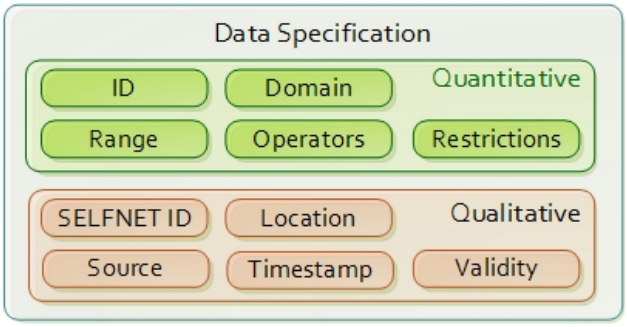

Figure 8.3: Example of data encapsulation

Reasoning, (1),(2),(3) with Projection and (7) with the administration of the use cases. Their tasks are summarized below.

- *Pattern Recognition.* identifies previously known or acquired patterns and regularities in facts related with aggregate data (i.e. $Fa(T_h), Fa(KPI), Fa(Ev)$), and returns Facts Fa with the results of their study. With this purpose, different internal tasks may be executed: study of the input data (both training data and samples to be analyzed), decision of the best suited data mining strategies for each context, feature extraction, construction of models/regressions, analysis of facts related with aggregate data in order to find and label verification. Note that the bibliography collects a plethora of pattern recognition methods, which are adapted to the needs of the use cases and to the characteristics of the different monitoring environments [DSBA16]. The SELFNET Analyzer focuses on two fundamental actions: the identification of signatures of previously known events [MLK14] and the detection of anomalies [Agg15].

- *Prediction component.* calculates the prediction metrics (as Facts Fa) associated with each use case from the observations provided by the aggregation stage (Thresholds T_H, Key Performance Indicators KPI and Events Ev). This implies different processing

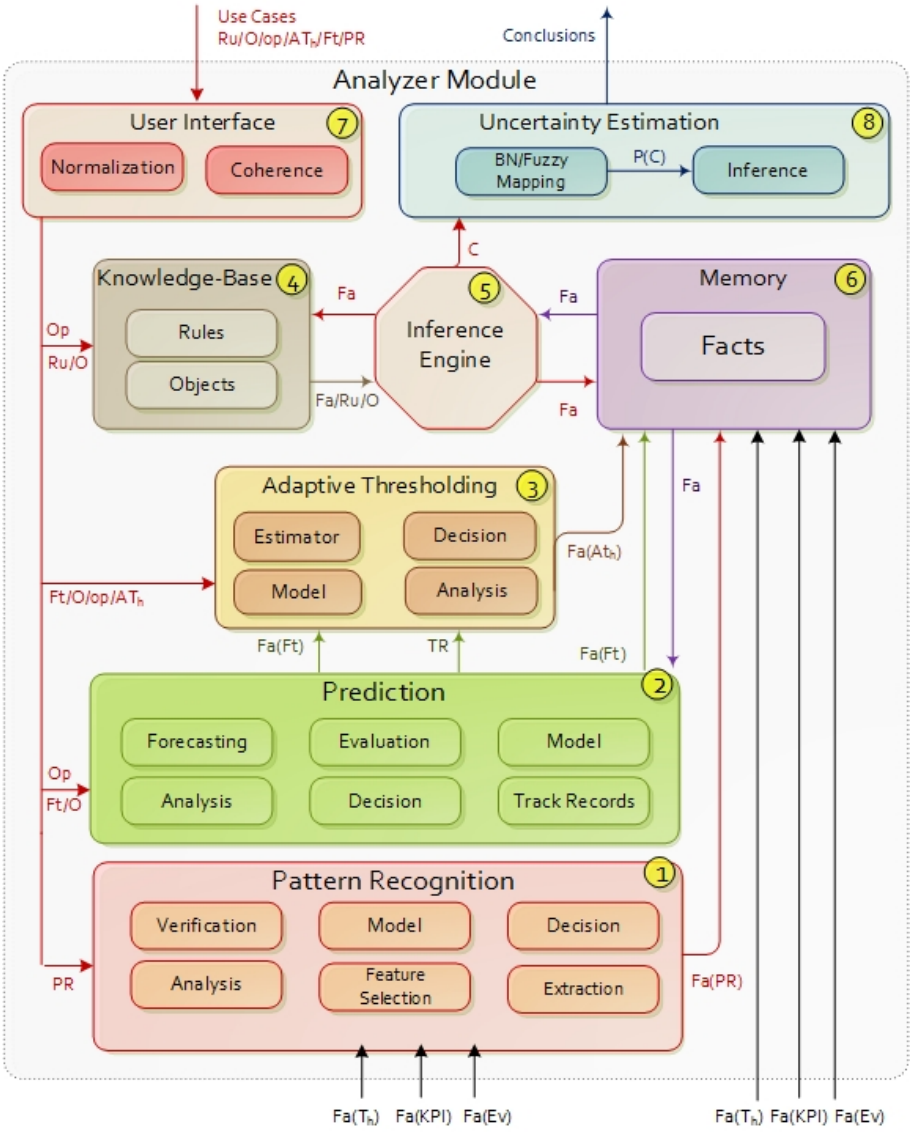

Figure 8.4: Analyzer module architecture

steps: management of a track record with the data required to build forecasting models, analysis of the data characteristics which are relevant for deciding the best suited prediction algorithms, construction of forecasting models, decision of prediction algorithms, forecasting and evaluation of the results in order to learn from the previous decisions. Note that as stated in [KD15], the prediction of network events enhances the optimization of resources, allows the deployment of proactive actions and anticipates risk identification. The SELFNET Analyzer focuses primarily on infer predictions from two data structures: time series and graphs. The first one aims to determine the evolution of the HoN metrics, hence it mainly implements exponential smoothing algorithms [GD80] and auto-regressive models [KHC+16]. On the other hand, the evolution of graphs is predicted in order to anticipate the discovery of new elements [BPC13] and facilitate the management of resources [RG12].

- *Adaptive thresholding.* establishes measures to approximate when the forecast errors must be taken into account when identifying symptoms. Therefore it receives as input parameters the values related with the prediction metrics (Track Record TR and Forecasts Ft), and returns adaptive thresholds AT_h. Their construction involves different steps, such as analyzing and extracting the main features from the input data, decision of the best suited thresholding algorithms, modeling and estimation of thresholds. The SELFNET Analyzer build adaptive thresholds from data represented as time series or graph, which allows inferring more accurate conclusions from every forecast generated by the prediction component. The main applicability of the adaptive thresholds is considering the context of the monitoring environment in the inference of new facts related with filtering [ZWH+16], and decreasing the false positives rates [BFK+17].

- *Knowledge-Base.* stores specific information about each use case. This data is represented by objects and rules. The objects O are the basic units of information (ex. temperature, congestion, latency, etc.). The rules Ru are the guidelines for reasoning that enable the inference of facts and conclusions. Facts, objects, and their values are interrelated through operations Op. A priori, in this approach online machine learning is not considered in order to acquire knowledge about the use cases in real-time [VE16], such as definition of new rules, prioritization, metric weighting, etc. (i.e. all information to be considered part of the original training and the specification of the use cases and their symptoms provided by operators).

- *Inference Engine.* applies rules Ru to the knowledge base in order to deduce new knowledge. This process would iterate as each new fact Fa in the knowledge base could trigger additional rules. Traditionally, inference engines operate in one of two modes: forward chaining and backward chaining [HRWL84]. The first initially considers previously known facts and infers new facts. On the other hand, backward chaining initially considers facts and tries to infer the causes that have led to them. Because the SELFNET Analyzer infers conclusions from discovered facts, the first approach is implemented. In addition, it is important to bear in mind that the easier implementation of the inference engine considers basic implication elimination rules (i.e. modus ponens rules) driven by propositional logic [MMRAT16]. They can be adapted to different representations of uncertainty, such as fuzzy logic [MF02], rough sets [CLL+15] or Bayesian networks [FN01]. But in order to facilitate the understanding of this proposal, the current specification of rules on the SELFNET Analyzer applies only basic propositional logic rules (as described in section 8.5), hence postponing a most complex but generic definition for future works.

- *Memory.* stores all the known facts Fa concerning with the use cases (ex. $Temperature = 3°$, $Latency > 200ms$, etc.) considering those predicted/inferred $(Fa(PR), Fa(AT_h), Fa(Ft))$ and those provided by the SELFNET Monitoring/Aggregation stages $(Fa(T_h), Fa(KPI), Fa(Ev))$. Metadata related with qualitative additional information about the nature of the discovered facts is also stored.

- *User interface.* configures Pattern Recognition PR for each use case and allows updating the knowledge-base by inserting, modifying or deleting data associated with every use case, such as objects O, rules Ru operations Op or prediction metrics Ft. The information is pre-processed aiming to ensure compatibility and coherence [Gil12]. The latter is particularly important, as it tries to avoid contradictions and ambiguity between rules, prior to their incorporation into the SELFNET intelligence.
- *Uncertainly Estimation.* complements the inference engine and facilitates the study of the conclusions bearing in mind their uncertainty. Its outputs are the acquired conclusions as potential symptoms of relevant incidences, their uncertainty and the information associated with their inference (facts, triggering rules, etc.). This is the only optional element of the architecture, since its use is only required when the SELFNET Diagnostic task [EN517] need to disambiguate conclusions, filter those of greater uncertainty or convert the logic on the Analyzer to data specified for upper layers of SELFNET. For example, when the inference engine operates on fuzzy logic rules, the element of Uncertainly Estimation generates a quantifiable result use-friendly for Diagnosis as crisp logic, given fuzzy sets and the corresponding membership degrees (i.e. defuzzification) [TC17].

8.3 Analyzer Inputs/Outputs

By studying the Analyzer Module as a black box model it is possible to focus more on its inputs/outputs and their relationship with the rest of the SELFNET components [NCC+16]. From this perspective, their information sources, nature of the data and behaviour in different circumstances are described. As shown in Figure. 8.5, the Analyzer Module depends on three sources of information. Two of them are external: the SELFNET Aggregation component and the use case operators; the last is internal: data generated by the Analyzer Module itself. The inferred conclusions are reported to the SELFNET Diagnosis Module as symptoms [EN517]. The role played by each of these elements is detailed below.

- *Aggregation.* Observations in SELFNET come to the Analyzer through the Aggregation Layer (Perception capabilities within the Endsley model). The information provided by this source contains facts concerning Events $Fa(Ev)$, Thresholds $Fa(T_H)$ and Key Performance Indicators $Fa(KPI)$ related to the current network status.
- *Use case operators.* The knowledge-base is specified from data acquired from the use case definitions. Because of this, use case operator may provide inference rules Ru (1), and declare the objects O (2), operations Op (3) and prediction metrics Ft (4) to be taken into account (i.e. what observations should be taken into account (2)?, how (3)? what data must be forecasted (4)? And how are they considered in order to acquire knowledge about the specific use case (1)?). Optionally, the use case operator may describe the adaptive thresholds AT_h to be calculated, and if pattern recognition PR is required, then configuring how it must be addressed.

- *Analyzer.* An important part of the information necessary for proper reasoning is generated by the Analyzer Module itself. It is gathered into a pair of groups: Perception and Machine Learning. The first block is imperative, and establishes facts Fa from pattern recognition $Fa(PR)$, forecasts $Fa(Ft)$ and adaptive thresholds $Fa(AT_h)$. On the other hand, machine learning may provide additional data to that provided by use case operators (definition of new rules Ru and description of prediction metrics Ft). Furthermore, it could generate information to improve the knowledge management (weight, prioritization, fusion, smoothing, etc.).
- *Diagnosis.* the final conclusions and symptoms that compose the SELFNET Situational Awareness are sent to Intelligent Diagnostic Module (Autonomic Management Sublayer) [EN517], via reports Re.

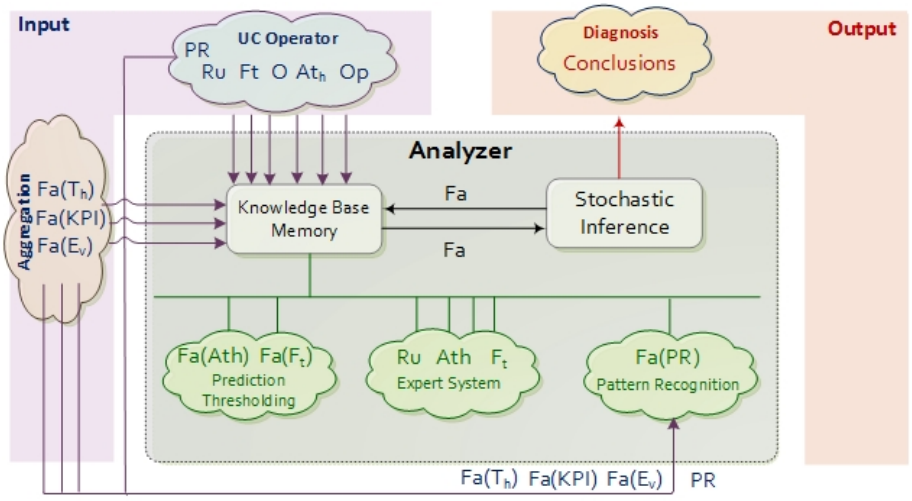

Figure 8.5: Analyzer module as a black box

8.4 Use Case Descriptors

This section describes the characteristics of Analyzer Module quantitative data and its categories. In Table 8.1 the quantitative data is summarized.

8.4.1 Object O

The objects $O = \{O_1, .., O_n\}$, $n \geq 1$ are definitions of the elements from which the system infers knowledge. They are added to the knowledge-base by use case operators. Their function is to describe the nature of the data in order to facilitate the selection of proper preprocessing and prediction methods. Objects are expressed as follows:

$$O_i : \{object \quad name \mid weight \mid noValues \mid range \quad of \quad values \quad Va\}$$

Examples:

$$\{Temperature \mid 1 \mid 1 \mid (-30°, 150°)\},$$
$$\{Link \quad Status \mid 0.7 \mid 1 \mid \{"Good", "Normal", "Bad"\}\},$$
$$\{Header \quad Encryption \mid 1.5 \mid 1 \mid (True, False)\},$$
$$\{Upper \quad Threshold \mid 2 \mid 1 \mid Y_t : t \epsilon T, \forall Y_i \epsilon R\}$$

Where *object name* acts as identification of the data category and the range of values limits the values that can be assigned. The *weight* is a field reserved for the future implementation of machine learning; it determines priority. Finally, $noValues$ anticipates its amount of possible values. Because of this, an object may be specified as a sequence of k previously defined objects or values interrelated. In this case, they are defined as follows.

$$O_i : \{object \quad name \mid weight \mid noValues \mid [Va_1][Va_2]...[Va_k]\}$$

Examples:

$$\{pairWeather \mid 1 \mid 2 \mid [temperature][humidity]\},$$
$$\{metricA \mid 2 \mid 4 \mid [TTL][lenght][port][ipAddress]\},$$
$$\{tSerieB \mid 2 \mid 8 \mid [R][R][R][R][R][R][R][R]\},$$

The specification of sequences of the same value repeated several times in a row can be simplified by the indicator $: i$, where i is the number of times it repeats. For example, the previous example:

$$\{tSerieB \mid 2 \mid 8 \mid [R][R][R][R][R][R][R][R]\},$$

It may be simplified as follows:

$$\{tSerieB \mid 2 \mid 8 \mid [R] : 8\},$$

8.4.2 Operations *Op*

The operations $Op = \{Op_1, .., Op_n\}$, $n \geq 1$ are definitions of binary relationships between facts Fa, objects O or their possible values Va. Initially, the knowledge-base provides a basic battery of operations (Ex. All arithmetic operations, propositional logic relationships, basic statistic expressions, etc.). When a use case is on-boarded, operators should declare the set of operations to be taken into account and their restrictions. This is achieved by the following layout:

$$Op_i : \{name \mid symbol \mid priority \mid operands \mid description\}$$

Examples:

$$\{Equal \mid = \mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\},$$
$$\{LGT \mid \geq \mid 1 \mid (Fa, O, Va) \geq (Fa, O, Va) \mid left \quad is \quad GE\},$$
$$\{And \mid \wedge \mid 1 \mid (Fa) \wedge (Fa) \mid logical \quad conjuction\},$$
$$\{Addition \mid + \mid 3 \mid (Fa, Vo) + (Fa, Vo) \mid addition\},$$

Where *name* refer to the identification of the operation in the predefined battery, *symbol* is its shortened representation, *priority* its position in the hierarchy of operations, *operands* limits the categories of operands applicable on each side of the binary expression, and *description* briefly explains its functionality in natural language.

8.4.3 Facts Fa

The facts $Fa = \{Fa_1, .., Fa_n\}$, $n \geq 1$ are the basic elements of the SELFNET reasoning. They are added to the memory of the Analyzer Module by the Aggregation layer or deduced by the inference engine. Facts must be accompanied by a *timestamp* indicating when they have been stated, the *location* on which they are valid and a *weight* that determine their priority. The location refers to SELFNET elements (ex. physical machines, virtual nodes, etc.). The priority is a field reserved by future machine learning weighting. *Uncertainty* describes its probability of being true. Facts are described as the following expression:

$$Fa_i : \{expresion \mid weight \mid uncertainty \mid timestamp \mid location\}$$

Example:

$$\begin{gathered}\{Ur_{threshold} = MaxValue \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid NodeA\},\\ \{Temperature \geq 80° \mid 0.7 \mid 0.98 \mid Today \quad 03:41:20 \mid VM15\},\\ \{KPI7 = UrTh + MaxT \mid 1.2 \mid 1 \mid Today \quad 03:41:20 \mid VM15\},\end{gathered}$$

8.4.4 Rules Ru

The $rules = \{Ru_1, .., Ru_n\}$, $n \geq 1$ describe how the Analyzer Module acquires new knowledge via rule-based expert system. In order to facilitate their specification, they are declared as propositional logic expressions, as follows:

$$\begin{gathered}(Fa(X) \vee Fa(Y)) \longrightarrow Fa(Z)\\ (Fa(X) \wedge Fa(Y)) \vee (Fa(X) \wedge \neg Fa(Z)) \longrightarrow Fa(B)\\ (Fa(C) \vee Fa(Y)) \vee \neg(Fa(A) \wedge Fa(Z)) \longrightarrow Fa(B)\end{gathered}$$

The rules are accompanied by the identification of the *use case* on which they are valid, and their *priority* of inference. Note that in order to enhance scalability, the rules of each use case are totally independent from the others. Rules are detailed as follows:

$$Rule : \{rule \mid priority \mid use \quad case\}$$

Examples:

$$\begin{gathered}\{(Fa(X) \vee Fa(Y)) \longrightarrow Fa(Z) \mid 1 \mid SP\}\\ \{(Fa(B)) \longrightarrow Fa(Y) \mid 2 \mid SO\}\\ \{(Fa(X) \wedge Fa(Y)) \vee (Fa(X) \wedge \neg Fa(Z)) \longrightarrow Fa(B) \mid 1 \mid SH\}\end{gathered}$$

8.4.5 Forecast Ft

The $Forecasts = \{Ft_1, .., Ft_n\}$, $n \geq 1$ are specifications of the objects that must be projected per use case. In this way it is possible to enhance the selection of prediction algorithms and forecasting models. Given the nature of the monitoring environment, *a priori*, this approach only considers predictions on two data types: time series and graphs. The time series allow estimating the evolution of KPI or thresholds from concrete

locations on SELFNET (physical infrastructure, network devices, virtualization, etc.). The prediction on graphs facilitates the inference of changes on large regions of the SELFNET topology, such as spreading of congestion, inclusion of new network elements or failures. This expert system considers prediction results as facts, so Ft only refers to their specification when on-boarding new use cases. Figure 8.4 predictions as facts are declared as $Fa(Ft)$. The following expression describes the forecasts on time series:

$$Ft_i : \{timeSeries \mid object \mid domain \quad lenght\}$$

Examples:

$$\{timeSeries \mid O_1 \mid obs \mid t+5\}$$
$$\{timeSeries|O_2 \mid time \mid Today13:28:15\}$$

Where *timeSeries* is a reserved word indicating that the prediction is on time series, *object* declares the nature of the data to be analyzed, and *domain* is the extension of the prediction. The examples show two reserved words, *obs* (observations) and *time* (timestamp). When the time is measured in observations, the length of the prediction is indicated from the initial time instant t and the amount of coming observations (ex. t+5 indicates forecast the next five observations). On the other hand, timestamps directly detail how long must be the prediction (ex. Today 13:28:15 indicates the requirement of forecast a certain object between now and 13:28:15 today). Note that the term *timeSeries* is used to describe the way in which data is structured and not the prediction algorithm. A record tracking of this nature could be forecasted by traditional time series methods (auto-regressive moving average, exponential smoothing, extrapolation, etc.) but also by other very different approaches (drifting, naive-based algorithms, *Artificial Neural Network* (ANN), *Support Vector Machines* (SVMs), etc.). It is up to the decision component of Prediction, select the most appropriate forecasting strategy. If the prediction considers observations on graphs, the forecasts are specified as follows:

$$Ft_i : \{graph \mid object \mid noVertex \mid domain \mid lenght\}$$

Examples:

$$\{graph \mid O_1 \mid 30 \mid obs \mid t+20\}$$
$$\{graph \mid O_2 \mid 45 \mid timestamp \mid Today19:12:07\}$$
$$\{graph \mid O_3 \mid 10 \mid timestamp \mid Today22:30:00\}$$

Where *graph* is the reserved word to declare predictions on graphs. *object* is the nature of the data on the edges of its incidence matrix. $noVertex$ is the number of vertex (i.e. dimension $noVertex - by - noVertex$ of its complete adjacency matrix). The last two parameters (*domain* and *length*) have the same function as in the expression of *timeSeries* prediction (indicate the measurement of time and the extension of the prediction).

8.4.6 Thresholds T_h

The thresholds $T_h = \{T_{h1}, .., T_{hn}\}$, $n \geq 1$ are specifications of fault tolerance limits related with values assigned to objects O. They are calculated by the SELFNET Aggregation task, but their specification is part of the use case operators. Thresholds are described as the following expression:

$$T_{hi} : T_h \quad name \mid object$$

Examples:

$$\{maxTemp \mid O(temperature)\}$$
$$\{maxConnections \mid O(nConnections)\}$$
$$\{minQuality \mid O(QoS)\}$$

Where T_h name is the threshold identification and *object* is the object on which it acts.

8.4.7 Adaptive Thresholds T_h

The adaptive thresholds $AT_h = \{AT_{h1}, .., AT_{hn}\}$, $n \geq 1$ are specification of fault tolerance limits related with values assigned to predictions Ft. They are calculated by the component of prediction of the Analyzer Module, but must be specified by the use case operators. Similarly to the Forecasts descriptions, initially they act on time series or graphs. They are described as follows:

$$AT_{hi} : AT_h \quad name \mid data \quad structure \mid CI \mid forecast$$

Examples:

$$\{maxTemp \mid timeSeries \mid 0.95 \mid Ft(A)\}$$
$$\{maxWorkload \mid graph \mid 0.90 \mid Ft(X)\}$$

Where AT_h *name* is the identification of the adaptive threshold, *data structure* is *timeSeries* or *graph* depending on the representation of the predicted data, CI is the confidence interval on which it is built by the Adaptive Thresholding component and forecast is the prediction from which it is created.

8.4.8 Pattern Recognition PR

The pattern recognition configurations $PR = PR_1, ..., PR_n$, $n \geq 1$ are specifications of how facts Fa related with aggregate data are analyzed in order to determine their similarity with previously established reference information. The outputs of pattern recognition actions are facts that display the degree of the similarity observed. Each PR action is defined as follows:

$$PR_i : \{PR \quad name \mid objectIn \mid ObjectOut \mid action \mid reference \quad data\}$$

Examples:

$$\{botnetTraffic \mid O(tFlow) \mid O(dist) \mid match \mid D(dataset1)\}$$
$$\{paylScan \mid O(payload) \mid O(dist) \mid anomaly \mid D(dataset2)\}$$
$$\{usrVerify \mid O(uAction) \mid O(dist) \mid anomaly \mid D(dataset3)\}$$

Where $PR \quad name$ is the action identificator, $objectIn$ is the nature of the data to be studied, $objectOut$ is the nature of the object recipient of the similarity degree, $action$ is the reserved word associated with the type of analysis to be performed. The default actions are "match" for matching observations with the reference data and "anomaly" for outlier detection. Finally, $referencedata$ is the identification of the dataset D to be taken into account.

8.4.9 Datasets D

The Datasets $D = \{D_1, ..., D_n\}$, $n \geq 1$ is the initial reference data to be required by pattern recognition actions. Given that Analyzer Module does not consider online training, all the reference data is provided by the use cases via User Interface. Datasets are declared as the following expression:

$$D_i : \{D \quad name \mid object \mid type \mid source\}$$

Examples:

$$\{legitimatePayload \mid O(payload) \mid model \mid Repository1\}$$
$$\{mySet1|O(flowMetrics) \mid collection \mid Repository2\}$$
$$\{autoreplicationGens \mid O(binary) \mid signature \mid Repository3\}$$

Where $D \quad name$ is the dataset identifier and $object$ is the nature of its samples. In this first approach, the dataset can be framed by three types: "collection", "model" or "signature". Firstly, "collection" refers to a set of raw observations directly extracted from the monitoring environment. On the other hand, "model" is a pre-processed description of the data to be analysed. Finally, "signature" indicates exactly patters to be identified. The field $source$ determines where the dataset is found (ex: path, url, repository, etc.).

8.4.10 Conclusions C

The conclusions $C = \{C_1, ..., C_n\}$, $n \geq 1$ are the subset of the group of facts Fa specified for a use case to be satisfied, that form part of the Situational Awareness of the network. When a conclusion is inferred, it is reported to the Diagnostic module [EN517] for being a potential indicator of situations. These symptoms are defined by use case operators as follows:

$$C_i : \{C \quad name \mid use \quad case \mid fact\}$$

Example:

$$\{gridlock \mid SP \mid Fa(A)\}$$
$$\{overHeating \mid SH \mid fa(X)\}$$

Where C name is the conclusion identificator, use case is the associated SELFNET use case, and fact is the triggering conclusion. Conclusions are reported to Diagnostic Module as follows:

$$Re_i : \{C \quad name \mid use \quad case \mid fact \mid uncertainty \mid trigger\}$$

Example:

$$\{gridlock \mid SP \mid Fa(A) \mid 0.85 \mid Fa(B), Fa(C), Ru(1)\}$$
$$\{overHeating \mid SH \mid fa(X) \mid 0.75 \mid Fa(x), Ru(3)\}$$

Where *uncertaintly* is the probability of being certain and trigger is the list of rules Ru or facts Fa that take part of its inference.

8.5 Examples of Specification and Workflows

This section describes three examples of data specification and workflows on the Analyzer Module.

8.5.1 UC 1: Device Temperature Analysis

This section describes an example of a sensor related with self-healing use case.

8.5.1.1 Description

The use case (*myTemp*) requires identifying symptoms related with overheat on network devices. This is a very basic example where prediction and adaptive thresholding are not considered. Therefore the decision thresholds are static and were built at Aggregation.

8.5.1.2 Initial Status

The Analyzer Module disposes of a battery of predefined operations, including basic arithmetic calculations, logical and statistical functions.

8.5.1.3 Use Case Specification

First, the use case operators specify the basic objects to be taken into account: the temperature of the devices and its upper threshold.

$$O_1 : \{Temperature \mid 1 \mid 1 \mid R\}$$
$$T_{h1} : \{maxTemp \mid O_1\}$$

Second is indicating the operators that are required and how they are taken into account:

$$Op_1 : \{Equal \mid = \mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$$
$$Op_2 : \{LGT \mid \geq \mid 1 \mid (Fa, O, Va) \geq (Fa, O, Va) \mid leftisGE\}$$

Third, the conclusions to be satisfied:

$$C_1 : \{overheat \mid myTemp \mid Fa(O_1) \geq Fa(T_{h1})\}$$

The last step is declaring the inference rules:

$$Ru_1 : \{Fa(O_1) \geq Fa(T_{h1}) \longrightarrow Fa(C_1) \mid 1 \mid myTemp\}$$

8.5.1.4 Workflow

At runtime, Aggregation layer notify to the Analyzer Module the facts related with myTemp use case. Some of them concern the temperature on SELFNET devices, for example:

$$\begin{gathered} Fa_1 : \{O_1 = 35° \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid NodeA\} \\ Fa_2 : \{O_1 = 76° \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid NodeB\} \\ \ldots \\ Fa_5 : \{O_1 = 80° \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid NodeB\} \end{gathered}$$

Note that uncertainty is 1 because the sensors are deterministic (100% probability of provide the correct temperature). The facts refer to the static thresholding are:

$$\begin{gathered} Fa_7 : \{T_{h1} = 79° \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid All\} \\ Fa_8 : \{T_{h1} = 79° \mid 1 \mid 1 \mid Today \quad 12:22:16 \mid All\} \end{gathered}$$

These facts are provided by Aggregation, and they are directly included on the memory of the Analyzer Module. If they are updated for the same location (ex. Fa_5 and Fa_6), the latest version is considered by the inference engine. After certain period of observation, the inference engine tries to deduct new knowledge from the rule-set of every use case. In myTemp, the Analyzer Module tries to infer conclusions for Ru_1. At $Today \quad 12:22:17$ the systems satisfy the first conclusion: $Fa_5(O_1 = 80°) \geq Fa_8(O_1 = 79°)$, so the fact $Fa(C_1)$ is added to memory:

$$Fa_9 : \{Fa_5 \geq Fa_8 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid NodeB\}$$

The location NodeB is considered because it is the more restrictive between NodeB,All. So the symptom C_1 has been discovered, and it is reported to Diagnostic Module as follows:

$$Re_1 : \{overheat \mid myTemp \mid Fa_9 \mid 1 \mid Fa_5, Fa_8, Ru_1\}$$

The inference engine will continue operating looking for new symptoms.

8.5.2 UC 2: Network Congestion Analysis

This section describes an example of a sensor related with self-optimization use case.

8.5.2.1 Description

The use case to be managed *Self-Congestion* (SC) requires identifying symptoms related with traffic congestion on SELFNET elements. In this example, prediction and adaptive thresholding are considered.

8.5.2.2 Initial Status

The Analyzer Module disposes of a battery of predefined operations, including basic arithmetic calculations, logical and statistical functions.

8.5.2.3 Use Case Specification

First, the use case operators specify the basic objects to be taken into account: the congestion level monitored and its prediction.

$$O_1 : \{congestion \mid 1 \mid 1 \mid [0,1]\}$$
$$Ft_1 : \{timeSeries \mid O_1 \mid obs \mid t+3\}$$

Next, they define an adaptive threshold to be automatically generated from the information provided by the record tracking and the Adaptive Thresholding.

$$AT_{h1} : \{maxCongestion \mid timeSeries \mid 0.95 \mid Ft_1\}$$

Second, it is specified what operators are required and how they are taken into account:

$$Op_1 : \{Equal \mid = \mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$$
$$Op_2 : \{LGT \mid \geq \mid 1 \mid (Fa, O, Va) \geq (Fa, O, Va) \mid leftisGE\}$$

Third, the conclusions are identified:

$$C_1 : \{gridlock \mid SC \mid Fa(O_1) \geq Fa(AT_{h1})\}$$

The last step is declaring the inference rules:

$$Ru_1 : \{Fa(O_1) \geq Fa(AT_{h1}) \longrightarrow Fa(C_1) \mid 1 \mid SC\}$$

8.5.2.4 Workflow

At runtime, Aggregation layer notify to the Analyzer Module facts related with the self-congestion use case, for example:

$$Fa_1 : \{O_1 = 0.6 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ServerA\}$$
$$Fa_2 : \{O_1 = 0.65 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ServerA\}$$
$$Fa_3 : \{O_1 = 0.65 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ServerA\}$$
$$Fa_4 : \{O_1 = 0.67 \mid 1 \mid 1 \mid Today \quad 12:22:21 \mid ServerA\}$$
$$Fa_5 : \{O_1 = 0.68 \mid 1 \mid 1 \mid Today \quad 12:22:24 \mid ServerA\}$$
$$....$$
$$Fa_{44} : \{O_1 = 0.66 \mid 1 \mid 1 \mid Today \quad 12:22:50 \mid ServerA\}$$
$$Fa_{45} : \{O_1 = 0.67 \mid 1 \mid 1 \mid Today \quad 12:22:52 \mid ServerA\}$$
$$Fa_{47} : \{O_1 = 0.69 \mid 1 \mid 1 \mid Today \quad 12:22:56 \mid ServerA\}$$
$$Fa_{48} : \{O_1 = 0.86 \mid 1 \mid 1 \mid Today \quad 12:22:58 \mid ServerA\}$$
$$Fa_{49} : \{O_1 = 0.97 \mid 1 \mid 1 \mid Today \quad 12:23:01 \mid ServerA\}$$

The construction of predictive models requires certain amount of previous observations; in this case, it considered the first 45 facts. They are handled by the record tracking in order to extract the needed information and define time series. At Today 12:22:52 (when Fa_{45} is deducted), the forecasting component provides the first prediction Ft_1 for the instant t+3. Then a new fact is included to the memory:

$$Fa_{46} : \{AT_{h1} = 90 \mid 1 \mid 1 \mid Today12:23:52 \mid ServerA\}$$

It triggers the rule Ru_1, because $Fa_49(O_1 = 0.97) \geq Fa_46(AT_{h1} = 90)$, and the conclusion C is satisfied. The new knowledge $Fa(C_1)$ is added to memory as:

$$Fa_{50} : \{Fa_{49} \geq Fa_{46} \mid 1 \mid 1 \mid Today12:23:01 \mid NodeB\}$$

Finally, the symptom is reported to Diagnostic Module as follows:

$$Re_1 : \{gridlock \mid SC \mid Fa_{50} \mid 1 \mid Fa_{49}, Fa_{46}, Ru_1\}$$

8.5.3 UC 3: Payload Analysis

This section describes an example of a sensor related with self-protection use case.

8.5.3.1 Description

This use case *Self-Guard* (SG) requires identifying symptoms related with anomalous payloads on SELFNET traffic. In this example, pattern recognition actions are considered.

8.5.3.2 Initial Status

The Analyzer Module disposes of a battery of predefined operations, including basic arithmetic calculations, logical and statistical functions. The external repositories (Rep1, Rep2) provide collection of Legitimate (Rep1) and malicious (Rep2) SELFNET traffic observations.

8.5.3.3 Use Case Specification

Firstly, the use case operators specify the basic objects to be taken into account; in this example they are the payload of the SELFNET traffic O_1, its similarity with the legitimate payload dataset O_2 and the malicious samples O_3.

$$O_1 : \{payload \mid 1 \mid 1 \mid hexadecimal\}$$
$$O_2 : \{simLegi \mid 1 \mid 1 \mid \{0..1\}\}$$
$$O_3 : \{simMal \mid 1 \mid 1 \mid \{0..1\}\}$$

Secondly, it is specified what operators are required and how they are taken into account:

$$Op_1 : \{Equal \mid = \mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$$
$$Op_2 : \{LT \mid > \mid 1 \mid (Fa, O, Va) > (Fa, O, Va) \mid leftisG\}$$

Next, they define the datasets to be taken into account.

$$D_{legi} : \{legitimatePayload \mid O(payload) \mid collection \mid Rep1\}$$
$$D_{mal} : \{maliciousPayload \mid O(payload) \mid collection \mid Rep2\}$$

And then the pattern recognition actions to be executed:

$$PR_1 : \{legMeasure \mid O_1 \mid O_2 \mid anomaly \mid D(D_{legi})\}$$
$$PR_2 : \{malMeasure \mid O_1 \mid O_3 \mid anomaly \mid D(D_{mal})\}$$

Conclusions are identified as follows:

$$C_1 : \{maliciousContent \mid SC \mid Fa(O_2) < Fa(O_3)\}$$

And the following rules are onboarded:

$$Ru_1 : \{Fa(O_2) < Fa(O_3) \longrightarrow Fa(C_1) \mid 1 \mid SP\}$$

8.5.3.4 Workflow

At runtime, Aggregation layer notifies to the Analyzer Module facts related with the SG use case:

$$Fa_1 : \{O_1 = FF217 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionA\}$$
$$Fa_2 : \{O_1 = FFFFF \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionB\}$$
$$Fa_3 : \{O_1 = 00DE8 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionA\}$$
$$Fa_4 : \{O_1 = A4FC9 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionA\}$$
$$Fa_5 : \{O_1 = FF218 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ConexionC\}$$
$$....$$
$$Fa_{38} : O_1 = F0279 \mid 1 \mid 1 \mid Today \quad 12:22:23 \mid ConexionA$$

Each time a new payload is observed, the SELFNET Analyzer Module performs the pattern recognition actions PR_1 and PR_2. This returns new facts:

$$Fa_{1R1} : \{O_2 = 0.9 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionA\}$$
$$Fa_{1R2} : \{O_3 = 0.2 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionA\}$$
$$Fa_{2R1} : \{O_2 = 0.9 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionB\}$$
$$Fa_{2R2} : \{O_3 = 0.18 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionB\}$$
$$Fa_{3R1} : \{O_2 = 0.8 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ConexionC\}$$
$$Fa_{3R2} : \{O_3 = 0.21 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ConexionC\}$$
$$...$$

At Today 12:22:23 the following facts are discovered:

$$Fa_{32R1} : \{O_2 = 0.66 \mid 1 \mid 1 \mid Today \quad 12:22:23 \mid ConexionA\}$$
$$Fa_{32R2} : \{O_3 = 0.92 \mid 1 \mid 1 \mid Today \quad 12:22:23 \mid ConexionA\}$$

They trigger the rule Ru_1, because $Fa_{32R1}(O_2 = 0.66) < Fa_{32R2}(O_3 = 0.92)$, and then the conclusion C is satisfied. The new knowledge $Fa(C_1)$ is added to memory as:

$$Fa_{50} : \{Fa_{32R1} < Fa_{32R1} \mid 1 \mid 1 \mid Today \quad 12:23:23 \mid ConexionA\}$$

Finally, the symptom is reported to Diagnostic Module as follows:

$$Re_1 : \{suspiciousPayload \mid SC \mid Fa_{50} \mid 1 \mid Fa_{32R1}, Fa_{32R2}, Ru_1\}$$

Table 8.1: Summary of UC data specification

Data	Category	Provider	Destination	Format
Object (simple) O	Specification	Use Case	Analyzer	O_i : {*object name* | *weight* | *noValues* | *range of values Va*}
Object (mult) O	Specification	Use Case	Analyzer	O_i : {*object name* | *weight* | *noValues* | $[Va_1][Va_2]...[Va_k]$}
Operation Op	Specification	Use Case	Analyzer	Op_i : {*name* | *symbol* | *priority* | *operands* | *description*}
Facts Fa	Assessment	Agg-Ana	Analyzer	Fa_i : {*expresion* | *weight* | *uncertainty* | *timestamp* | *location*}
Rule Ru	Specification	Use Case	Analyzer	*Rule* : {*rule* | *priority* | *use case*}
Forecast (ts) Ft	Specification	Use Case	Analyzer	Ft_i : {*timeSeries* | *object* | *domain lenght*}
Forecast (G) Ft	Specification	Use Case	Analyzer	Ft_i : {*graph* | *object* | *noVertex* | *domain* | *lenght*}
Threshold T_h	Specification	Use Case	Analyzer	T_{hi} : {T_h *name* | *object*}
A. Threshold At_h	Specification	Use Case	Analyzer	AT_{hi} : {AT_h *name* | *data structure* | *CI* | *forecast*}
Datasets D	Specification	Use Case	Analyzer	D_i : {D *name* | *object* | *type* | *source*}
Pattern Recognition	Specification	Use Case	Analyzer	PR_i : {PR *name* | *objectIn* | *ObjectOut* | *action* | *reference data*}
Conclusion C	Specification	Use Case	Analyzer	C_i : {C *name* | *use case* | *fact*}
Report Re	Report	Analyzer	Diagnosis	Re_i : {C *name* | *use case* | *fact* | *uncertainty* | *trigger*}

8.6 SELFNET Analyzer Orchestrator

In order to manage all the information (input/output) and analytical functions this section describes the design and specification of the SELFNET Analyzer orchestrator. It poses an important challenge in terms of configurability, synchronization and resource management. The SELFNET Analyzer relationship with the rest of the project components is detailed in section 8.3. There two main information sources as facts Fa: i) Aggregation ($Fa(Ev)$, Thresholds $Fa(T_h)$ and $Fa(KPI)$) and ii) Internal analytic elements (pattern recognition $Fa(PR)$, forecasts $Fa(Ft)$ and adaptive thresholds $Fa(AT_h)$). The final conclusions that compose the SELFNET Situational Awareness are sent to Diagnosis module labeled as symptoms, via reports. When initiated, the SELFNET Analyzer is a tabula rasa without actions nor reasoning to be orchestrated. If a new use case is onboarded, the SELFNET Analyzer Orchestrator starts to work, as is described in this section.

8.6.1 Orchestration Considerations

This section describes the general considerations and terms to understand the orchestration of the SELFNET Analyzer Module.

- *Symptoms and events.* The Diagnosis layer of SELFNET distinguishes two groups of reports: symptoms and events. The first one contains conclusions generated through analytics. On the other hand, events are signals on which it is not necessary to carry

out actions related to Artificial Intelligence, such as pattern recognition, prediction or logical inference. In the remainder of this work, it is assumed that the expression $Fa(Ev)$ strictly refers to aggregated metrics extracted from the monitored events, instead of the event themselves. For example, alerts issued by the IDS involved in the use case self-protection are, by definition, events. Given their relevance, they must be directly addressed to the Diagnosis layer, so it is not possible to assume the cost in time that involves the execution of complex analytical calculations on them. However, it is possible to generate metrics that facilitate the making of future decisions or even new alerts (not in real-time). For example, the Aggregation layer may provide information about the number of alerts per observation, mean, variance, emission intervals, among others. From this information stronger conclusions could be inferred.

- *Rule based Inference.* Given the SELFNET framework and the nature of the monitored data, the decision to implement a rule-based inference engine as a symptom discovery tool brings many benefits, among them: i) Rule engines allow to use case administrators decide "What to do", not "How to do it'. Because of this, it makes it easy to express solutions to difficult problems and specify the onboard of future use cases. ii) It brings logic and data separation, where data is in the domain of objects, and the logic is in the rules Ru. iii) It provides centralization of the knowledge required to infer symptoms. iv) Rule-based systems are fast and scalable: some algorithms (ex. RETE, Leaps, Treat, etc.) [BV97] and their optimizations [GC12] provide very efficient ways of matching rule patterns to the use cases domain object data. These are especially efficient when facts change in small portions as the rule engine can remember past matches. For example, this happens with the information periodically provided by a particular SELFNET sensor. But rule-based systems also pose drawbacks: the first of them is high dependency of the rule set. If the rules are not consistent, coherent or reasonably specific, the results obtained will probably not be as expected [LP95]. On the other hand, they are susceptible to bad practices. For example, rule-based systems allow storing, managing and updating rules as data. It is common that they are mistakenly used to generate new rules or even update them at runtime, which is out of the scope of these technologies. Finally, it is important to bear in mind that the scalability of rule-based systems has a negative impact in terms of resource consumption. In this regard, it is worth mentioning the consequences of their two most frequent ways to scale [WH92]: firstly, if the number of facts is acceptable, but the number of rules is very high, there will be an important increase in the computation time of their processing. On the opposite, if the number of facts is very high, but the number of rules is acceptable, a larger amount of memory is required for storage. Note that if the number of inputs and rules are large, then both, memory and efficiency are penalized. In the context of SELFNET it is expected to receive a large number of facts, but operate on small rule sets. Consequently, it is expected that the scalability of the expert rule-based system will lead to the use of a greater amount of storage space.

- *Data granularity.* SELFNET is a complex monitoring scenario where a large amount

of sensors collect information about the state of the network in real time. This information is processed in the aggregation layer, which provides the necessary metrics to acquire knowledge. For this purpose, the Analyzer must perform complex calculations. As will be described in the later sections, aggregated data will not be raw processed. Instead, it will be packed as *Aggregated Data Bundle* (ADB) which will periodically be loaded by the Analyzer and converted into facts. Each ADB is the summary of the aggregated metrics calculated in a time interval T translated into facts Fa. It can therefore be stated that ADB may be abstracted as an observation on a time series of records that facilitate the network awareness. It is assumed that the effectiveness and performance of the analytic depends on the T, and how representative the information on the ADB is. Note that a priori, the data within an ADB does not overlap the metrics on other ADBs (this aspect could be revised later for future optimizations). For example, let the time series $Y = \{Y_t : t \in T\}$ where $\{Y_1, Y_2, ..., Y_k, \}$ $k = 7$ assuming the construction of ADBs on $T = 1$, the SELFNET Analyzer will sequentially deal with 7 ADBs, i.e. $\{ADB_1, ADB_2, ..., ADB_k, \}$ (see Figure. 8.6). Through the use of this strategy a massive and continuous input of information is avoided, which facilitate the initialization of the implemented data mining algorithms. Likewise, the information is managed and processed in an orderly manner, which also reduces the number of inconsistencies between the new facts and the data stored in the working memory. Finally, as is illustrated at the next section, the deployment of optimization method based on the exploitation of concurrence is facilitated.

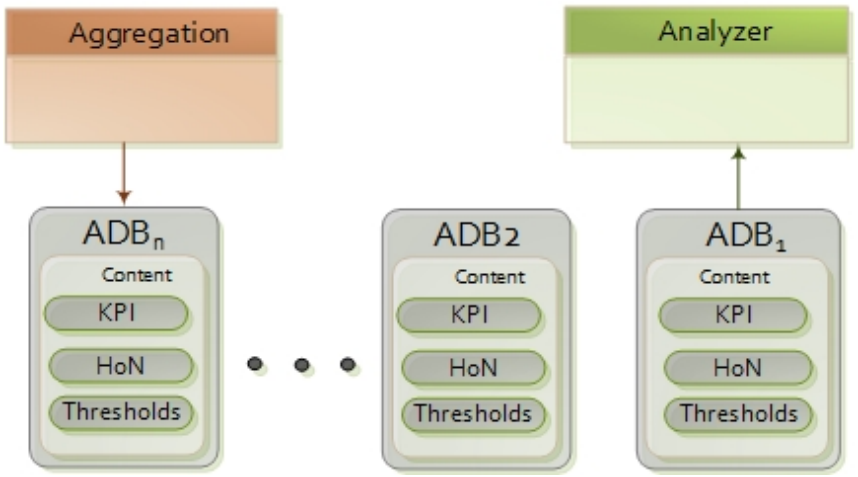

Figure 8.6: Communication by ADBs

- *Persistence.* The SELFNET Analyzer does not provide persistence of the data loaded as ADBs. The monitored raw data and aggregated metrics are conveniently stored in the Big Data platform located at the Aggregation layer. Facts Fa not implicated in prediction/pattern recognition are discarded once their ADB is completely processed and the conclusions are inferred. This means that, in this case, facts Fa are temporally stored in a local short-term memory only for the duration of their analysis. On the other hand, facts Fa required for prediction/pattern recognition may temporally persist throughout the analysis of various ADBs. This is because they compose the time series

and graphs needed to build models/regressions. Note that these data structures have limited size, which once reached involves eliminating the more obsolete observations via *First In First Out* (FIFO) policies. Once an ADB is completely analyzed and the conclusions are reported to the Diagnosis layer as symptoms, the working memory of the rule-based expert systems is restarted. Only the necessary facts for the construction of the time series and graphs are temporarily conserved, but this is outside the working memory. When loading a new ADB, facts on time series and graphs are again, added to the working memory as $(Fa(T_h), Fa(Ft)$ and $Fa(PR))$.

- *Analytic pipelining.* Analytics are executed as a linear pipeline of sets of data processing elements connected in series, where the output of an input is the input of the next one. When an ADB reach the SELFNET Analyzer, a sequence of processing elements is executed, where intelligence actions (i.e. logic inference, pattern recognition, prediction) and preprocessing steps (load ADBs, data encapsulation, generation of reports) are chronologically separated, and their inputs/outputs are shared by buffer storage structures. So it is possible to state that this first approach considers a buffered-synchronous pipeline analytic architecture. Its main advantages are: great organization of information to process, mitigation of inconsistencies between the new facts and the data being analyzed, easy of design and modularity. These aspects allows managing every set of actions independently, which facilitates debugging, troubleshooting tasks and provide a more accurate assessment of the performance of their analytic actions. But it is important to bear in mind that defining this scheme also poses several challenges, among them try sets of actions of similar complexity in order to enable optimization strategies based on parallelism, the fact that the delay in a task may slow down the execution of those that depend on it, and in the case of implement parallelism, the best suited politics of temporal memory sharing must be identified.

8.6.2 SELFNET Analyzer Orchestration Steps

The SELFNET Analyzer orchestration is separated into seven main steps: use case Onboarding $[O]$, Discovery $[DIS]$, Patter Recognition $[PR]$, Prediction $[FT]$, Adaptive Thresholding $[AT_h]$, Knowledge inference $[KI]$ and Notification $[N]$. They are illustrated in Figure. 8.7 and described in detail bellow.

- *Onboarding [O].* The onboarding step is executed only once per use case. It corresponds to the component User Interface, and allows updating the knowledge-base by inserting, modifying or deleting data associated with every use case, such as objects O, rules Ru operations Op or prediction metrics Ft. When a new use case is onboarded, the input data is normalized, and in order to avoid runtime errors, the coherence of the new specification is validated. Then the Analyzer is prepared to accommodate the new operations, hence including the specified information on the existing data structures, memory allocation and synchronization of the onboarded actions with the previous loaded configurations.

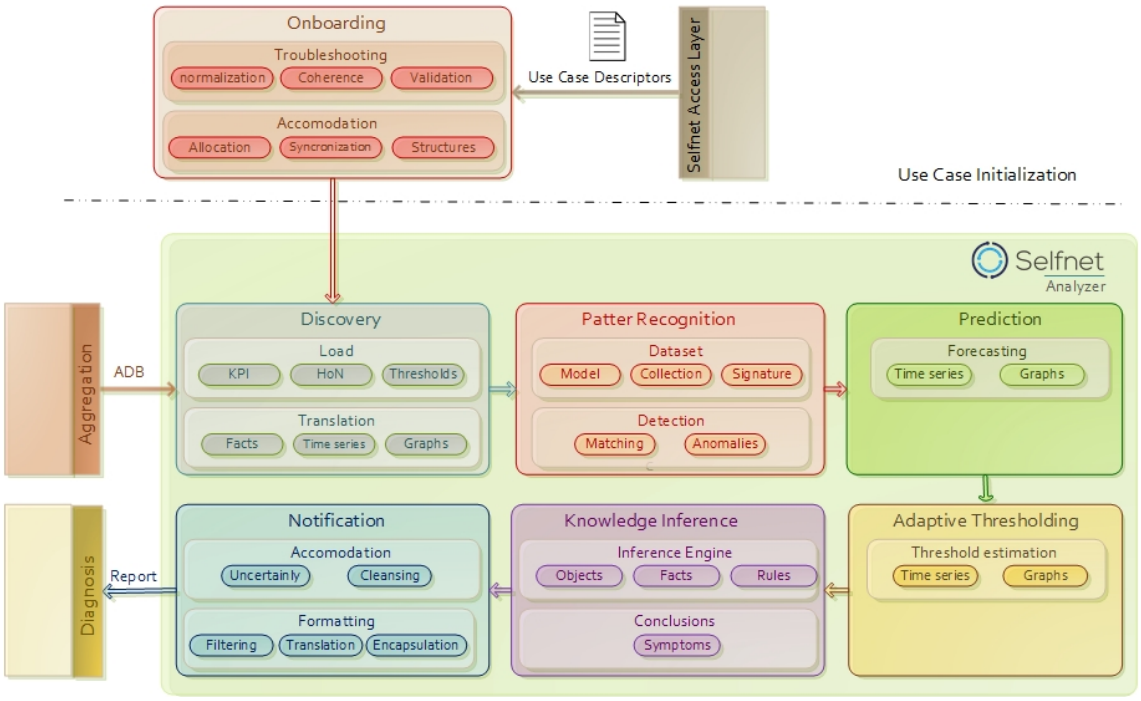

Figure 8.7: Sets of actions on the analyzer

- *Discovery [DIS].* The discovery step is the link between the SELFNET Aggregation and Analyzer layers. These tasks periodically receive ADB which summarize the SELFNET aggregated observations. From the loaded KPI, events and thresholds, the Analyzer build facts $\mathrm{F}a(KPI)$, $\mathrm{F}a(Ev)$ and $\mathrm{F}a(AT_h)$). If they are required for prediction, patter recognition or adaptive thresholding, the Analyzer includes these observations in the temporally stored time series or graphs. Note that independent facts are removed at the end of the ADB processing, as well as the new knowledge acquired from them.

- *Pattern Recognition [PR].* The set of actions related with pattern recognition implies the access to the datasets with models, sample collection or signatures, and the detection of matches or outliers. The acquired facts may be considered by prediction, pattern recognition or adaptive thresholding, as well as to infer knowledge on the rule-based expert system.

- *Prediction [FT].* The set of actions related with prediction include the construction of forecasting models/regression, the decision of the best suited algorithms by considering the nature of the input data, and the estimation of its evolution. As is the case on the pattern recognition activities, the generated facts may be considered to infer knowledge on the rule-based expert system, and also to identify adaptive thresholds.

- *Adaptive Thresholding [ATH].* This set of operations establishes measures to approximate when the forecasting errors must be taken into account when identifying symptoms. In order to enhance the information reported to the Diagnosis layer, the new facts are provided to the rule-based expert system, hence contributing to the

inference of new knowledge.

- *Knowledge inference [KI].* This step executes the tasks related with the rule-based expert system. It considers the data provided by the sources of information mentioned above, among them facts directly built from aggregated data, pattern recognition, prediction and adaptive thresholding steps. The acquired knowledge is included in the SELFNET Analyzer working memory. Conclusions are transmitted to the notification capabilities as potential symptoms.

- *Notification [N].* The set of actions on Notification corresponds to those on the component Uncertainty Estimation at the original SELFNET Analyzer architecture. They are the link between the SELFNET Diagnosis layer and the knowledge acquired by the Analyzer. This step performs two main groups of tasks: accommodation and formatting. The first one filter redundant and low representative information. Once the ADB is completely analyzed, these actions erase and restart the auxiliary functionalities on the analytics and the several data structures; only the information required for build time series and graphs from data included in future ADBs is temporally persistent. On the other hand, the group of actions related with formatting, translate internal information of the analyzer to crisp data handy by Diagnosis. Then it is reported.

8.6.3 Analyzer Orchestration Example

The illustrative use case to be managed Self-QoSOverwatch (SQoS) report symptoms related with suspicious QoS decreasing. If a QoS variation is observed in the latest observations, a new fact is added. In this context, prediction and adaptive thresholding are considered, and it is assumed that the forecasting algorithm requires at least n=8 observations for building the prediction model. Table 8.2 shows its onboarding descriptors.

Table 8.2: Self-QoSOverwatch specification

Item	Descriptor
Object	$O_1 : \{QoS \quad decrement \mid 1 \mid 1 \mid [0,1]\}$
Forecast	$Ft_1 : \{Time \quad series \mid O_1 \mid obs \mid t+1\}$
Adaptive Threshold	$AT_{h1} : \{maxQoS \quad decrement \mid timeSeries \mid 0.95 \mid Ft_1\}$
Operator	$Op_1 : \{Equal \mid = \mid 1 \mid (Fa,O,Va) = (Fa,O,Va) \mid equal\}$
Operator	$Op_2 : \{LFT \mid \geq \mid 1 \mid (Fa,O,Va) \geq (Fa,O,Va) \mid left \quad is \quad GE\}$
Conclusion	$C_1 : \{Suspicius \quad QoS \quad variation \mid SQoS \mid Fa(O_1) \geq Fa(AT_{h1})\}$
Rule	$Ru_1 : \{Fa(O_1) \geq Fa(AT_{h1}) \longrightarrow Fa(C_1) \mid 1 \mid SC\}$

In this example different ADBs are loaded and analyzed, as follows.

1) The descriptors of the SQoS use case are loaded by the SELFNET Analyzer. Then, the memory for storing temporal containers of objects O_1, forecasts Ft_1, adaptive thresholds AT_{h1} and facts Fa is allocated. Prediction capabilities on time series are required, so the

data structures to support time series are initiated.

2) The ADB_1 with aggregated instances of O_1 is requested to the Aggregation layer and then it is processed. The SELFNET Analyzer built the following facts from the information gathered by the network elements (NodeA, NodeB, NodeC, NodeD):

$$\begin{aligned}
&Fa(O)[idO1] : \{O_1 = 0.60|1|1|Today \quad 12:22:15|NodeA\}\\
&Fa(O)[idO1] : \{O_1 = 0.60|1|1|Today \quad 12:22:15|NodeA\}\\
&Fa(O)[idO2] : \{O_1 = 0.65|1|1|Today \quad 12:22:15|NodeB\}\\
&Fa(O)[idO3] : \{O_1 = 0.61|1|1|Today \quad 12:22:15|NodeC\}\\
&Fa(O)[idO4] : \{O_1 = 0.62|1|1|Today \quad 12:22:15|NodeD\}
\end{aligned}$$

3) Given that the Analyzer does not dispose of time series of n=8 facts per sensor, prediction is not possible. Hence, adaptive thresholding is not performed. Because there are not facts related with adaptive thresholds, the rule Ru_1 where $Fa(O_1) \geq Fa(AT_{h1}) \longrightarrow Fa(C_1)$ cannot be triggered. So conclusions related with symptoms are not notified to the diagnosis layer. On the other hand, given that the acquired facts are related with time series analysis (i.e. prediction and adaptive thresholding), they cannot be deleted before loading the following ADBs, but the rule-based inference engine is reinitiated.

4) The Analyzer performs the same actions (Step 2 and 3) from ADB_2 until ADB_7. Table 8.3 shows the facts built for these set of ADBs.

Table 8.3: Facts ADB_2 to ADB_7

ADB	Fact
ADB_2	$Fa(O)[idO5] : \{O_1 = 0.63\|1\|1\|Today \quad 12:23:15\|NodeA\}$ $Fa(O)[idO6] : \{O_1 = 0.64\|1\|1\|Today \quad 12:23:15\|NodeB\}$ $Fa(O)[idO7] : \{O_1 = 0.65\|1\|1\|Today \quad 12:23:15\|NodeC\}$ $Fa(O)[idO8] : \{O_1 = 0.66\|1\|1\|Today \quad 12:23:15\|NodeD\}$
ADB_3	$Fa(O)[idO9] : \{O_1 = 0.62\|1\|1\|Today \quad 12:24:15\|NodeA\}$ $Fa(O)[idO10] : \{O_1 = 0.70\|1\|1\|Today \quad 12:24:15\|NodeB\}$ $Fa(O)[idO11] : \{O_1 = 0.72\|1\|1\|Today \quad 12:24:15\|NodeC\}$ $Fa(O)[idO12] : \{O_1 = 0.63\|1\|1\|Today \quad 12:24:15\|NodeD\}$
ADB_4	$Fa(O)[idO13] : \{O_1 = 0.60\|1\|1\|Today \quad 12:25:15\|NodeA\}$ $Fa(O)[idO14] : \{O_1 = 0.72\|1\|1\|Today \quad 12:25:15\|NodeB\}$ $Fa(O)[idO15] : \{O_1 = 0.73\|1\|1\|Today \quad 12:25:15\|NodeC\}$ $Fa(O)[idO16] : \{O_1 = 0.65\|1\|1\|Today \quad 12:25:15\|NodeD\}$
ADB_5	$Fa(O)[idO17] : \{O_1 = 0.62\|1\|1\|Today \quad 12:26:15\|NodeA\}$ $Fa(O)[idO18] : \{O_1 = 0.71\|1\|1\|Today \quad 12:26:15\|NodeB\}$ $Fa(O)[idO19] : \{O_1 = 0.76\|1\|1\|Today \quad 12:26:15\|NodeC\}$ $Fa(O)[idO20] : \{O_1 = 0.63\|1\|1\|Today \quad 12:26:15\|NodeD\}$
ADB_6	$Fa(O)[idO21] : \{O_1 = 0.63\|1\|1\|Today \quad 12:27:15\|NodeA\}$ $Fa(O)[idO22] : \{O_1 = 0.70\|1\|1\|Today \quad 12:27:15\|NodeB\}$ $Fa(O)[idO23] : \{O_1 = 0.71\|1\|1\|Today \quad 12:27:15\|NodeC\}$ $Fa(O)[idO24] : \{O_1 = 0.60\|1\|1\|Today \quad 12:27:15\|NodeD\}$
ADB_7	$Fa(O)[idO25] : \{O_1 = 0.61\|1\|1\|Today \quad 12:28:15\|NodeA\}$ $Fa(O)[idO26] : \{O_1 = 0.72\|1\|1\|Today \quad 12:28:15\|NodeB\}$ $Fa(O)[idO27] : \{O_1 = 0.73\|1\|1\|Today \quad 12:28:15\|NodeC\}$ $Fa(O)[idO28] : \{O_1 = 0.62\|1\|1\|Today \quad 12:28:15\|NodeD\}$

5) The ADB_8 with aggregated instances of O_1 is requested to the Aggregation layer and then it is processed. The SELFNET Analyzer built the following facts from the information gathered by the network elements:

$$Fa(O)[idO29] : \{O_1 = 0.60|1|1|Today \quad 12:29:15|NodeA\}$$
$$Fa(O)[idO30] : \{O_1 = 0.73|1|1|Today \quad 12:29:15|NodeB\}$$
$$Fa(O)[idO31] : \{O_1 = 0.72|1|1|Today \quad 12:29:15|NodeC\}$$
$$Fa(O)[idO32] : \{O_1 = 0.64|1|1|Today \quad 12:29:15|NodeD\}$$

6) At this point, there are n=8 facts per sensor in the time series to be predicted, so the forecasting method are able to estimate the next observation (t+1) as specified in the use case definition. The temporally stored data is summarized in Table 8.4.

Table 8.4: Summary of information on time series at SQoS

Time	N	NodeA	NodeB	NodeC	NodeD
12:22:15	1	0.60	0.65	0.61	0.62
12:23:15	2	0.63	0.64	0.65	0.66
12:24:15	3	0.62	0.70	0.72	0.63
12:25:15	4	0.6	0.72	0.73	0.65
12:26:15	5	0.62	0.71	0.76	0.63
12:27:15	6	0.63	0.7	0.71	0.6
12:28:15	7	0.61	0.72	0.73	0.62
12:29:15	8	0.6	0.73	0.72	0.64
Forecast	n+1	0.61	0.72	0.72	0.63

The following facts related with prediction are acquired:

$$Fa(Ft)[idF1] : \{Ft_1 = 0.61|1|1|Today \quad 12:29:15|NodeA\}$$
$$Fa(Ft)[idF2] : \{Ft_1 = 0.72|1|1|Today \quad 12:29:15|NodeB\}$$
$$Fa(Ft)[idF3] : \{Ft_1 = 0.72|1|1|Today \quad 12:29:15|NodeC\}$$
$$Fa(Ft)[idF4] : \{Ft_1 = 0.63|1|1|Today \quad 12:29:15|NodeD\}$$

7) The following facts related with the adaptive thresholds built from the predictions are acquired:

$$Fa(Ath)[idA1] : \{Ath_1 = 0.62|1|1|Today \quad 12:29:15|NodeA\}$$
$$Fa(Ath)[idA2] : \{Ath_1 = 0.73|1|1|Today \quad 12:29:15|NodeB\}$$
$$Fa(Ath)[idA3] : \{Ath_1 = 0.74|1|1|Today \quad 12:29:15|NodeC\}$$
$$Fa(Ath)[idA4] : \{Ath_1 = 0.64|1|1|Today \quad 12:29:15|NodeD\}$$

8) The recent calculated thresholds are not applicable to the current observations, so the rule Ru_1 where $Fa(O_1) \geq Fa(AT_{h1}) \longrightarrow Fa(C_1)$ cannot be triggered. Conclusions related with symptoms are not notified to the diagnosis layer. Note that for the observation i only the predictions and adaptive thresholds calculated at $0, ..., i-1$ can be considered; stated in another way: predictions and adaptive thresholds calculated at i are only valid for the next $i+1$ observations, when it can be verified whether they have been fulfilled.

9) The ADB_9 with aggregated instances of O_1 is requested to the Aggregation layer and then it is processed. The SELFNET Analyzer built the following facts from the information gathered by the network elements (NodeA, NodeB, NodeC, NodeD):

$$\begin{aligned}
&Fa(O)[idO33] : \{O_1 = 0.60|1|1|Today \quad 12:30:15|NodeA\}\\
&Fa(O)[idO34] : \{O_1 = 0.82|1|1|Today \quad 12:30:15|NodeB\}\\
&Fa(O)[idO35] : \{O_1 = 0.62|1|1|Today \quad 12:30:15|NodeC\}\\
&Fa(O)[idO36] : \{O_1 = 0.60|1|1|Today \quad 12:30:15|NodeD\}
\end{aligned}$$

10) Not pattern recognition actions are declared.

11) The following facts about predictions for the next observations are calculated:

$$\begin{aligned}
&Fa(Ft)[idF5] : \{Ft_1 = 0.60|1|1|Today \quad 12:30:15|NodeA\}\\
&Fa(Ft)[idF6] : \{Ft_1 = 0.75|1|1|Today \quad 12:30:15|NodeB\}\\
&Fa(Ft)[idF7] : \{Ft_1 = 0.72|1|1|Today \quad 12:30:15|NodeC\}\\
&Fa(Ft)[idF8] : \{Ft_1 = 0.62|1|1|Today \quad 12:30:15|NodeD\}
\end{aligned}$$

12) New facts related with adaptive thresholds are calculated:

$$\begin{aligned}
&Fa(Ath)[idA5] : \{Ath_1 = 0.61|1|1|Today \quad 12:30:15|NodeA\}\\
&Fa(Ath)[idA6] : \{Ath_1 = 0.76|1|1|Today \quad 12:30:15|NodeB\}\\
&Fa(Ath)[idA7] : \{Ath_1 = 0.73|1|1|Today \quad 12:30:15|NodeC\}\\
&Fa(Ath)[idA8] : \{Ath_1 = 0.63|1|1|Today \quad 12:30:15|NodeD\}
\end{aligned}$$

13) The rule-based inference engine processes the discovered facts by applying the rule Ru_1. Given that the condition $Fa(O_1) \geq Fa(AT_{h1})$ is satisfied for the data gathered by NodeB, The following fact related with Self-QoSOverwatch is inferred.

$$Fa[idF1] : \{Fa(idO34) \geq Fa(idA2)|1|1|Today \quad 12:30:15)|NodeB\}$$

Which describes the conclusion C_1:

$$C_1[idC1] : \{SuspiciousQoSvariation|SQoS|Fa(idO34) \geq Fa(idA2)\}$$

14) The following symptom is reported to the Diagnosis layer:

$$Re_1[idR1] : \{SuspiciousQoSvariation|SQoS|idF1|1|idF1, idA2, idO34, Ru_1\}$$

In figure 8.8 the specification of a use case (XML format) is shown.

```
<!-- Note: the order of the descriptors is relevant
<myTemp>
    <DesObject>
        <name>temperature</name>
        <weight>1</weight>
        <noValues>1</noValues>
        <range>1..N</range>
    </DesObject>
    <DesOperator>
        <name>equals</name>
        <symbol>=</symbol>
        <priority>1.0</priority>
        <listLeftOperands>
        <!-- Fact, Object or Value-->
            <category>Object</category>
        </listLeftOperands>
        <listRightOperands>
        <!-- Fact, Object or Value-->
            <category>Value</category>
        </listRightOperands>
    </DesOperator>
    <DesOperator>
        <name>LGT</name>
        <symbol>></symbol>
        <priority>1.0</priority>
        <listLeftOperands>
        <!-- Fact, Object or Value-->
            <category>Fact</category>
        </listLeftOperands>
        <listRightOperands>
        <!-- Fact, Object or Value-->
            <category>Fact</category>
        </listRightOperands>
    </DesOperator>
    <DesThreshold>
        <name>maxTemperature</name>
        <object>temperature</object>
    </DesThreshold>
    <DesFact>
        <DesFact>
    <DesFact>
    <DesConclusion>
    <DesRule>
</myTemp>
```

Figure 8.8: XML example of use case descriptor

8.7 Summary

In this chapter, the application of analysis and intelligence in 5G networks was explained. We introduced the design of SELFNET Analyzer Module and its data specification. Our design provides pattern recognition, reasoning and prediction capabilities to infer the possible symptoms, facilitating the diagnosis and decision-making tasks. The main contribution of SELFNET Analyzer Module is its general, simple and scalable approach, allowing new rules and metrics in the analysis process when a new use case is added by the operator. SELFNET sensors gather information from several data sources such as virtual elements, LTE, SDN and traditional network devices; and thus the gathered information can be subject of analysis. Furthermore, this proposal was built to support new analytic capabilities by means of a plugin based approach.

Chapter 9

Conclusions and Future Works

In general terms, analysis and intelligence capabilities play an important role in addressing 5G requirements, in combination with key-enabled technologies such as SDN, NFV, cloud computing, etc. All of these domains can take advantage of forecasting, pattern recognition, artificial intelligence and advanced analysis concepts. In this way, 5G networks expect to provide enhanced capacities related to the network management and the detection of possible harmful problems. For its part, the diagnosis of data is required in order to know what the real cause of the event is. Because of this, the intelligence is provided in two phases: i) analysis stage and ii) decision-making; similar to a medical evaluation, where firstly the symptoms are detected and then based on it a treatment is applied.

The research results presented in this thesis aid to cover 5G intelligent needs by means the introduction of data analysis and situational awareness concept. This proposal takes into account SDN and NFV applied to this kind of systems. Firstly, this work proposes a generalized architecture for 5G incidence management based on the combination of the cognitive model for situational awareness proposed by Endsley and traditional risk management guidelines and standards. In this way, the automation of proactive and reactive deployment of countermeasures is facilitated. The proposed architecture also takes advantage of every 5G data source in order to consider the quality of the context in the decision-making process.

As stated at the beginning of this document, this thesis is part of SELFNET project which provides a smart autonomic management framework for 5G mobile networks. SELFNET enables the autonomic deployment of SDN/NFV functions and the reconfiguration of network devices and service parameters in order to mitigate existing or potential problems, without affecting the SLAs or the the quality of service provided to end users. One of the main challenges of SELFNET is to analyze the data from five data sources and provide a scalable and a use-case driven approach able to support different kind of rules, analysis functionalities and heterogeneous data. In this context, this work proposes the SELFNET Analyzer framework which is based on the situational awareness concept.

SELFNET Analyzer framework provides a general purpose scheme easily adapted to the operator needs and it is able to overcome the design constraints of current

monitoring environments. SELFNET Analyzer Framework is able to analyse information from heterogeneous sources such as SDN elements, virtual devices or metrics from specialized sensors. Another important characteristic is that the SELFNET Analyzer facilitates the incorporation of different analysis strategies, such as novel prediction or pattern recognition algorithms. This framework is also able to operate indistinctly with very different data mining and machine learning paradigms, among them conventional information, big data or high dimensional data. The use of any of them does not imply design changes, being simply an implementation problem. As a result, this proposal is adaptable to future technologies and approaches.

The proposed data specification to accommodate new use cases is simple and adjustable. Note that SELFNET implements a triad of services: self-protection, self-healing and self-optimization with completely different features and dependences (metrics, network devices to be monitored, prediction/pattern recognition algorithms, etc.). It is important to emphasize that loading new use cases is based only on configuration changes, without the need to modify the Analyzer implementation or to include additional software. However, the effectiveness of the SELFNET Analyzer Framework depends on the quality of the specification inserted by operators when configuring the analyzer functionalities.

In summary, SELFNET Analyzer Module provides pattern recognition, reasoning and prediction capabilities to infer the possible symptoms, facilitating the diagnosis and decision-making tasks. The main contribution of SELFNET Analyzer Module is its general, simple and scalable approach, allowing new rules and metrics in the analysis process when a new use case is added by the operator. Furthermore, this proposal was built to support new analytic capabilities by means a plugin based approach and taking into account the situation awareness concept. This work also proposes the data specification to define the data inputs to be considered in the diagnosis process.

9.1 Future Works

Bearing in mind that the introduction of data analysis and intelligent in 5G networks is at an early stage and the assumptions of SELFNET Analyzer framework, some future works can be conducted.

- Firstly, SELFNET Analyzer Framework only considers the diagnosis stage to facilitate the decision-making process, which is considered part of future work. Based on the symptoms provided by the Analyzer, the decision-making phase perform reactive or proactive responses, thus closing the intelligent loop required by SELFNET.

- This proposal does not take into account a countermeasure tracking system or feedback from the decision-making phase. Thus, this aspect will be covered in future work.

- This proposal is built over the assumption that the normal observations to be analysed match with the reference sample of the learning process. The stationary

environment proposed by SELFNET Analyzer brings a simple and efficient solution, but it is prone to slight failures when changes happen. Thus, the introduction of mechanisms to work in non-stationary monitoring environments is also part of the ongoing work.

- SELFNET Analyzer is not able to deal with complex stationary monitoring environments, where the quality of the analytics decrease with time. Given the importance of this kind of scenarios, this is an aspect that must be studied in future work.

- SELFNET is a complex monitoring scenario where a large amount of sensors collect information from different data sources in real time. The Analyzer performs complex calculations and the data will be periodically packed, loaded and converted into facts. In this regard, another aspect of interest is identifying quality indicators related with the granularity of the information contained in the "Aggregated Packet" ADB. From them, it is possible to improve the effectiveness of the analytic actions.

- Another important aspect to bear in mind is the execution pipeline of the Analyzer components in order to provide consistence and facilitate the organization of the received information. Thus, the investigation of methods to process and analyse the received information is also part of the ongoing work.

- It is important to note that the integration and testing of the proposed Analyzer Framework is part of the ongoing work that will be carry out over a real testbed in the last year of SELFNET project. In order to provide intelligence and self-management capabilities, there are some pending tasks to be done or completed not only in the Analyzer module but also in others SELFNET components. For instance, the Aggregation task and the three main use cases have been developed in parallel with the Analyzer Framework, and thus they are mutually dependent among them. Firstly, SELFNET Analyzer Module will retrieve real metrics from Aggregation sublayer. Secondly, in order to acquire reference samples through the training mode, real monitoring data from each use case is needed. Furthermore, the use case specification shall be onboarded by the use case operator. All of these integration tasks are part of the future works.

- At present some submodules of Analyzer framework are being implemented and tested in order to cover specific requirements of current use cases and to provide a wide battery of algorithms for future cases. This task involves the decision of which pattern recognition and forecasting methods best fit the requirements of SELFNET, how Analyzer Module is able to decide the best strategy for each data category and their integration into the system.

Bibliography

[5G 17a] 5G Americas. http://www.5gamericas.org/es/, April 2017.

[5G 17b] 5G Forum. http://www.5gforum.org/, April 2017.

[5GM17] Fifth Generation Mobile Communication Promotion Forum (5GMF). http://5gmf.jp/en, April 2017.

[AAKS98] D. Alexander, W. Arbaugh, A. Keromytis, and J. Smith. A Secure Active Network Environment Architecture: Realization in SwitchWare. *IEEE Network*, 12(3):37–45, May 1998.

[ABC+14] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang. What will 5G Be? *IEEE Journal on selected areas in communications*, 32(6):1065–1082, June 2014.

[ABR16] M. S. K. Awan, P. Burnap, and O. Rana. Identifying Cyber Risk Hotspots: A Framework for Measuring Temporal Variance in Computer Network Risk. *Computers & Security*, 57:31–46, 2016.

[Ada93] Eugene C Adam. Fighter Cockpits of the Future. In *Digital Avionics Systems Conference, 1993. 12th DASC., AIAA/IEEE*, pages 318–323, 1993.

[Agg15] C. C. Aggarwal. Outlier Analysis. In *Data Mining*, pages 237–263, 2015.

[AGHE+15] N. Amit, A. Gordon, N. Har El, M. Ben-Yehuda, A. Landau, A. Schuster, and D. Tsafrir. Bare-metal performance for virtual machines with exitless interrupts. *Communications of the ACM*, 59(1):108–116, 2015.

[AHGZ16] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati. Network Function Virtualization in 5G. *IEEE Communications Magazine*, 54(4):84–91, 2016.

[AHNR16] K. Apajalahti, E. Hyvönen, J. Niiranen, and V. Räisänen. Stare: Statistical Reasoning Tool for 5G Network Management. In *International Semantic Web Conference*, pages 21–25, 2016.

[AIS+14] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour. Design Considerations for a 5G Network Architecture. *IEEE Communications Magazine*, 52(11):65–75, November 2014.

[AR14] M. N. Alenezi and M. J. Reed. Uniform DoS Traceback. *Computers & Security*, 45:17–26, 2014.

[ARS16] M. Agiwal, A. Roy, and N. Saxena. Next Generation 5G Wireless Networks: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 18(3):1617–1655, 2016.

[Ave16] T. Aven. Risk Assessment and Risk Management: Review of recent Advances on their Foundation. *European Journal of Operational Research*, 253(1):1–13, 2016.

[BASS11] T. Benson, A. Akella, A. Shaikh, and S. Sahu. CloudNaaS: a Cloud Networking Platform for Enterprise Applications. In *Proceedings of the 2nd ACM Symposium on Cloud Computing*, page 8, 2011.

[BAX+16] T. S. Buda, H. Assem, L. Xu, D. Raz, U. Margolin, E. Rosensweig, D. R. Lopez, M.-I. Corici, M. Smirnov, and R. Mullins. Can Machine Learning aid in Delivering New Use Cases and Scenarios in 5G? In *Proceedings of the Network Operations and Management Symposium 2016*, pages 1279–1284, Istanbul, Turkey, April 2016.

[BBS14] C. Bouveyron and C. Brunet-Saumard. Model-based Clustering of High-dimensional Data: A Review. *Computational Statistics & Data Analysis*, 71:52–78, 2014.

[BFK+17] F. Boem, R. Ferrari, C. Keliris, T. Parisini, and M. M. Polycarpou. A Distributed Networked Approach for Fault Detection of Large-scale Systems. *IEEE Transactions on Automatic Control*, 62(1):18–33, 2017.

[BGH+14] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O Connor, P. Radoslavov, and W. Snow. ONOS: Towards an Open, Distributed SDN OS. In *Proceedings of the third workshop on Hot topics in software defined networking*, pages 1–6, 2014.

[BGMB14] N. Baldo, L. Giupponi, and J. Mangues-Bafalluy. Big Data Empowered Self Organized Networks. In *Proceedings of the 20th European Wireless Conference*, pages 1–8, Barcelona, Spain, May 2014.

[BHL+14] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski. Five Disruptive Technology Directions for 5G. *IEEE Communications Magazine*, 52(2):74–80, February 2014.

[BLMVGV17a] L. I. Barona López, J. Maestre Vidal, and L. J. García Villalba. An Approach to Data Analysis in 5G Networks. *Entropy*, 19(2):1–23, February 2017.

[BLMVGV17b] L. I. Barona López, J. Maestre Vidal, and L. J. García Villalba. Orchestration of Use-Case Driven Analytics in 5G Scenarios. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–25, April 2017.

[BLMVVC+16] L. I Barona López, J. Maestre Vidal, A. L. Valdivieso Caraguay, M.A. Sotelo Monge, and L. J. García Villalba. Seguridad en Redes de Nueva Generación: Arquitectura para la Gestión de Incidencias. pages 252–258, October 2016.

[BLVCGV15] L. I Barona López, A. L. Valdivieso Caraguay, and L. J. García Villalba. Extending OpenFlow in Virtual Networks. pages 252–258, 2015.

[BLVCGVL15] L. I Barona López, A. L. Valdivieso Caraguay, L. J. García Villalba, and D. López. Trends on Virtualisation with Software Defined Networking and Network Function Virtualisation. *IET Networks*, 4(5):255–263, 2015.

[BLVCMV+17] L. I. Barona López, A. L. Valdivieso Caraguay, J. Maestre Vidal, M. A. Sotelo Monge, and L. J. García Villalba. Towards Incidence Management in 5G Based on Situational Awareness. *Future Internet*, 9(3):1–14, January 2017.

[BLVCSMGV16] L. I. Barona López, A. L. Valdivieso Caraguay, M. C. Sotelo Monge, and L. J. García Villalba. Key Technologies in the Context of Future Networks: Operational and Management Requirements. *Future Internet*, 9(1):1–15, October 2016.

[BMMR+15] A. Belmonte Martin, L. Marinos, E. Rekleitis, G. Spanoudakis, and N. Petroulakis. Threat Landscape and Good Practice Guide for Software Defined Networks/5G. 2015.

[BPC13] M. Berlingerio, F. Pinelli, and F. Calabrese. Abacus: Frequent Pattern Mining-based Community Discovery in Multidimensional Networks. *Data Mining and Knowledge Discovery*, 27(3):294–320, 2013.

[BRL+14] J. Blendin, J. Rückert, N. Leymann, G. Schyguda, and D. Hausheer. Position Paper: Software-defined Network Service Chaining. In *2014 Third European Workshop on Software Defined Networks*, pages 109–114, 2014.

[BTAS14] B. Bangerter, S. Talwar, R. Arefi, and K. Stewart. Networks and Devices for the 5G Era. *IEEE Communications Magazine*, 52(2):90–96, February 2014.

[BV97] N. Bassiliades and I. Vlahavas. Processing Production Rules in DEVICE, an Active Knowledge Base System. *Data & Knowledge Engineering*, 24(2):117–155, November 1997.

[Cal99] K. Calvert. Architectural Framework for Active Networks Version 1.0, 1999.

[CB10] N.M. Chowdhury and R. Boutaba. A Survey of Network Virtualization. *Computer Networks*, 54(5):862–876, 2010.

[CBB+16] Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, and K. Stoddart. A Review of Cyber Security Risk Assessment Methods for SCADA Systems. *Computers & Security*, 56:1–27, 2016.

[CCF+05] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. Van der Merwe. Design and Implementation of a Routing Control Platform. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design and Implementation*, volume 2, pages 15–28, Berkeley, CA, USA, May 2005.

[CDLL15] L. M. Contreras, P. Doolan, H. Lønsethagen, and D. R. López. Operational, Organizational and Business Challenges for Network Operators in the Context of SDN and NFV. *Computer Networks*, 92:211–217, December 2015.

[CFP+07] M. Casado, M. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker. Ethane: Taking Control of the Enterprise. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 1–12, Stockholm, Sweden, August 2007.

[CLL+15] H. Chen, T. Li, C. Luo, S.-J. Horng, and G. Wang. A Decision-theoretic Rough Set Approach for Dynamic Data Mining. *IEEE Transactions on Fuzzy Systems*, 23(6):1958–1970, 2015.

[CNP03] A. Courtney, H. Nilsson, and J. Peterson. The Yampa Arcade. In *Proceedings of the ACM Workshop on Haskell*, pages 7–18, New York, NY, USA, August 2003.

[CRB09] N.M. K. Chowdhury, M. R. Rahman, and R. Boutaba. Virtual Network Embedding with Coordinated Node and Link Mapping. In *INFOCOM 2009, IEEE*, pages 783–791, 2009.

[cro17] EU CROWD Project; Connectivity Management for eneRgy Optimised Wireless Dense networks. Project reference: 318115. Funded under: FP7-ICT. http://www.ict-crowd.eu/, April 2017.

[CSD15] M. M. Chatzimichailidou, N. A. Stanton, and I. M. Dokas. The Concept of Risk Situation Awareness Provision: Towards a New Approach for Assessing the DSA about the Threats and Vulnerabilities of Complex Socio-technical Systems. *Safety Science*, 79:126–138, 2015.

[CZA+15] S. Chen, J. Zhao, M. Ai, D. Liu, and Y. Peng. Virtual RATs and a Flexible and Tailored Radio Access Network Evolving to 5G. *IEEE Communications Magazine*, 53(6):52–58, 2015.

[DAKM15] N. Dahal, O. Abuomar, R. King, and V. Madani. Event Stream Processing for Improved Situational Awareness in the Smart Grid. *Expert Systems with Applications*, 42(20):6853–6863, 2015.

[DGK+13] P. Demestichas, A. Georgakopoulos, D. Karvounas, K. Tsagkaris, V. Stavroulaki, J. Lu, C. Xiong, and J. Yao. 5G on the Horizon: Key Challenges for the Radio-access Network. *IEEE Vehicular Technology Magazine*, 8(3):47–53, July 2013.

[dHyAP12] Ministerio de Hacienda y Administraciones Públicas. MAGERIT: Risk Analysis and Management Methodology for Information Systems. 2012.

[Dot15] P. Doty. US Homeland Security and Risk Assessment. *Government Information Quarterly*, 32(3):342–352, 2015.

[DRAP15] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar. Learning in Nonstationary Environments: A Survey. *IEEE Computational Intelligence Magazine*, 10(4):12–25, 2015.

[DSBA16] M. De Sanctis, I. Bisio, and G. Araniti. Data Mining Algorithms for Communication Networks Control: Concepts, Survey and Guidelines. *IEEE Network*, 30(1):24–29, 2016.

[EHE15] R. El-Hattachi and J. Erfanian. Next Generation of Mobile Networks. White paper, NGMN Alliance, February 2015.

[EN1] 5G Ensure. Deliverable D 2.3, Risk Assessment, Mitigation and Requirements (Draft). http://www.5gensure.eu/deliverables.

[EN517] SELFNET Consortium. Deliverable 5.3: Report and Prototypical Implementation of the Integration of the Algorithms and Techniques Used to Provide Intelligence to the Decision-Making Framework. 2016. https://selfnet-5g.eu/deliverables/, April 2017.

[End88] M. R. Endsley. Design and Evaluation for Situation Awareness Enhancement. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 32, pages 97–101, 1988.

[Eri13] D. Erickson. The Beacon Openflow Controller. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot topics in Software Defined Networking*, pages 13–18, New York, NY, USA, August 2013.

[ESHC98] M. R. Endsley, S. J. Selcon, T. D. Hardiman, and D. G. Croft. A Comparative Analysis of SAGAT and SART for Evaluations of Situation Awareness. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 42, pages 82–86, 1998.

[ETS] ETSI Industry Specification Group (ISG). Network Function Virtualization (NFV) use cases.

[ETS13a] ETSI Industry Specification Group (ISG). Network Function Virtualization (NFV) Architectural Framework, 2013.

[ETS13b] ETSI Industry Specification Group (ISG). Network Functions Virtualization White Paper, 2013.

[ETS14] ETSI Industry Specification Group (ISG). Management and Orchestration (ETSI MANO), 2014.

[FB14] U. Franke and J. Brynielsson. Cyber Situational Awareness–a Systematic Review of the Literature. *Computers & Security*, 46:18–31, 2014.

[FBMP12] P. Fonseca, R. Bennesby, E. Mota, and E. Passito. A Replication Component for Resilient OpenFlow-based Networking. In *Proceedings of the 2012 IEEE Network Operations and Management Symposium*, pages 933–939, Maui, HI, USA, April 2012.

[FGR+13] N. Foster, A. Guha, M. Reitblatt, A. Story, M. Freedman, N. Katta, C. Monsanto, J. Reich, J. Rexford, C. Schlesinger, D. Walker, and R. Harrison. Languages for Software Defined Networks. *IEEE Communications Magazine*, 51(2):128–134, February 2013.

[FHF+11] N. Foster, R. Harrison, M. Freedman, C. Monsanto, J. Rexford, A. Story, and D. Walker. Frenetic: A Network Programming Language. In *Proceedings of the 16th ACM international Conference on Functional Programming*, volume 46, pages 279–291, New York, NY, USA, September 2011.

[flo17] Floodlight. http://www.projectfloodlight.org/, April 2017.

[FN01] N. Fenton and M. Neil. Making Decisions: using Bayesian Nets and MCDA. *Knowledge-Based Systems*, 14(7):307–325, 2001.

[FRZ14] N. Feamster, J. Rexford, and E. Zegura. The Road to SDN: An Intellectual History of Programmable Networks. *ACM SIGCOMM Computer Communication Review*, 44(2):87–98, 2014.

[fStIEC05] International Organization for Standardization and the International Electrotechnical Commission. ISO/IEC 27002: Information Technology, Security Techniques, Code of Practice for Information Security Management. 2005.

[GC12] S. Guillaume and B. Charnomordic. Fuzzy Inference Systems: An integrated Modeling Environment for Collaboration between Expert Knowledge and Data using FisPro. *Expert Systems with Applications*, 39(10):8744–8755, August 2012.

[GD80] E. S. Gardner and D. G. Dannenbring. Forecasting with Exponential Smoothing: Some Guidelines for Model Selection. *Decision Sciences*, 11(2):370–383, 1980.

[GEB+13] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race. Towards Network-wide QoE Fairness Using Openflow-assisted Adaptive Video Streaming. In *Proceedings of the 2013 ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking*, pages 15–20, Hong Kong, China, August 2013.

[GHM+05] A. Greenberg, G. Hjaimtysson, D. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang. A Clean Slate 4D Approach to Network Control and Management. *ACM SIGCOMM Computer Communication Review*, 35(5):41–54, October 2005.

[Gil12] A. Gilio. Generalizing Inference Rules in a Coherence-based Probabilistic Default Reasoning. *International Journal of Approximate Reasoning*, 53(3):413–434, 2012.

[GJ15] A. Gupta and R. K. Jha. A Survey of 5G Network: Architecture and Emerging Technologies. *IEEE access*, 3:1206–1232, July 2015.

[GKP+08] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. NOX: Towards an Operating System for Networks. *ACM SIGCOMM Computer Communication Review*, 38(3):105–110, July 2008.

[GMUJ16] J. Garay, J. Matias, J. Unzilla, and E. Jacob. Service Description in the NFV Revolution: Trends, Challenges and a Way Forward. *IEEE Communications Magazine*, 54(3):68–74, 2016.

[GRF13] A. Guha, M. Reitblatt, and N. Foster. Machine verified Network Controllers. In *ACM SIGPLAN NOTICES*, volume 48, pages 483–494, June 2013.

[GVVCBL15] L. J. García Villalba, A. L. Valdivieso Caraguay, and L. I. Barona López. Use Cases Definition and Requirements of the System and its Components. *SELFNET Project*, 2015.

[GVVCBL16] L. J. García Villalba, A. L. Valdivieso Caraguay, , and L. I. Barona López. Report and Prototypical Implementation of the Monitoring and Discovery Module. *SELFNET Project*, 2016.

[HA14] S. Hansson and T. Aven. Is Risk Analysis Scientific? *Risk Analysis*, 34(7):1173–1183, 2014.

[HHB14] F. Hu, Q. Hao, and K. Bao. A Survey on Software-defined Network and OpenFlow: From Concept to Implementation. *IEEE Communications Surveys & Tutorials*, 16(4):2181–2206, May 2014.

[HHG16] R. Heijungs, P. Henriksson, and J. B. Guinée. Measures of Difference and Significance in the Era of Computer Simulations, Meta-Analysis, and Big Data. *Entropy*, 18(10):361, 2016.

[HNS+09] J.-C. Hourcade, Y. Neuvo, R. Saracco, W. Wahlster, and R. Posch. Future Internet 2020: Visions of an Industry Expert Group. Panel report, May 2009.

[Hol93] R. C. Holte. Very Simple Classification Rules Perform well on Most Commonly used Datasets. *Machine Learning*, 11(1):63–90, 1993.

[Hol14] E. Hollnagel. Is Safety a Subject for Science? *Safety Science*, 67:21–24, 2014.

[HP15] J. Halpern and C. Pignataro. Service Function Chaining (SFC) Architecture. Technical report, 2015.

[HRWL84] F. Hayes-Roth, D. Waterman, and D. Lenat. Building Expert Systems. 1984.

[HSMA14] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal. NFV: State of the Art, Challenges, and Implementation in Next Generation Mobile Networks (vEPC). *IEEE Network*, 28(6):18–26, 2014.

[IMT17] IMT-2020 (5G) Promotion Group. http://www.imt-2020.cn/en/introduction, April 2017.

[Int89] International Organization for Standardization and the International Electrotechnical Commission. Information Processing Systems Open Systems Interconnection Basic Reference Model-Part 2: Security Architecture. https://www.iso.org/standard/14256.html, February 1989.

[IZAD14] A. Imran, A. Zoha, and A. Abu-Dayya. Challenges in 5G: How to Empower SON with Big Data for Enabling 5G. *IEEE Network*, 28(6):27–33, 2014.

[JP13] R. Jain and S Paul. Network Virtualization and Software Defined Networking for Cloud Computing: a Survey. *IEEE Communications Magazine*, 51(11):24–31, 2013.

[KD15] C. Katris and S. Daskalaki. Comparing Forecasting Approaches for Internet Traffic. *Expert Systems with Applications*, 42(21):8172–8183, 2015.

[KF13] H. Kim and N. Feamster. Improving Network Management with Software Defined Networking. *IEEE Communications Magazine*, 51(2):114–119, February 2013.

[KFG15] D. King, A. Farrel, and N. Georgalas. The Role of SDN and NFV for Flexible Optical Networks: Current Status, Challenges and Opportunities. In *17th International Conference on Transparent Optical Networks (ICTON)*, pages 1–6, 2015.

[KHC+16] F. Kadri, F. Harrou, S. Chaabane, Y. Sun, and C. Tahon. Seasonal ARMA-based SPC Charts for Anomaly Detection: Application to Emergency Department Systems. *Neurocomputing*, 173:2102–2114, 2016.

[KK13] Shashi Kiran and Gary Kinghorn. Cisco Open Network Environment: Bring the Network Closer to Applications, 2013.

[KPH+15] W. Knowles, D. Prince, D. Hutchison, J. F. P. Disso, and K. Jones. A Survey of Cyber Security Management in Industrial Control Systems. *International Journal of Critical Infrastructure Protection*, 9:52–80, 2015.

[KRV+15] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig. Software-defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, 103(1):14–76, December 2015.

[KS16] N. P. Kuruvatti and H. D. Schotten. Framework to Support Mobility Context Awareness in Cellular Networks. In *Vehicular Technology Conference (VTC Spring), 2016 IEEE 83rd*, pages 1–5, 2016.

[KSC+11] H. Kim, S. Sundaresan, M. Chetty, N. Feamster, and W. K. Edwards. Communicating with Caps: Managing Usage Caps in Home Networks. In *Proceedings of the ACM SIGCOMM Conference*, pages 470–471, New York, NY, USA, August 2011.

[KSKD+12] A. Kassler, L. Skorin-Kapov, O. Dobrijevic, M. Matijasevic, and P. Dely. Towards QoE-driven Multimedia Service Negotiation and Path Optimization with Software Defined Networking. In *Proceedings of the 20th International Conference on Software, Telecommunications and Computer Networks*, volume 1, pages 1–5, Split, Croatia, September 2012.

[KT217] Open Platform for NFV (OPNFV). https://www.opnfv.org/, April 2017.

[KT317a] EU CONTENT Project; Convergence of Wireless Optical Network and iT rEsources iN SupporT of Cloud Services. Project reference: 318514 . Funded under: FP7-ICT. http://cordis.europa.eu/project/rcn/106689_en.html, April 2017.

[kt317b] EU METIS-II Project; Mobile and Wireless Communications Enablers for Twenty-Twenty (2020) Information Society-II. Project reference: 671680 . Funded under: H2020-ICT-2014-2. https://5g-ppp.eu/metis-ii/, April 2017.

[KT417a] EU CHARISMA Project; Converged Heterogeneous Advanced 5G Cloud-RAN Architecture for Intelligent and Secure Media Access. Project Reference: 671704. Funded under: H2020-ICT-2014-2. http://www.charisma5g.eu/, April 2017.

[KT417b] EU Flex5Gware Project. Flexible and Efficient Hardware/Software Platforms for 5G Network Elements and Devices. Project Reference: 671563. Funded under: H2020-ICT-2014-2. http://www.flex5gware.eu/, April 2017.

[KT417c] EU SONATA Project. Service Programing and Orchestration for Virtualized Software Networks. Project Reference: 671517. Funded under: H2020-ICT-2014-2. http://www.sonata-nfv.eu/, April 2017.

[LCMP12] P. Le Callet, S. Möller, and A Perkis. Qualinet White Paper on Definitions of Quality of Experience. *European Network on Quality of Experience in Multimedia Systems and Services*, 3, March 2012.

[LM15] Y.B Leau and S. Manickam. Network Security Situation Prediction: A Review and Discussion. In *International Conference on Soft Computing, Intelligence Systems, and Information Technology*, pages 424–435, 2015.

[LNR+04] T. Lakshman, T. Nandagopal, R. Ramjee, K. Sabnani, and T. Woo. The SoftRouter Architecture. In *Proceedings of ACM SIGCOMM Workshop on Hot Topics in Networking*, pages 1–6, New York, NY, USA, November 2004.

[LP95] A. Lunardhi and K.M. Passino. Verification of Qualitative Properties of rule-based Expert Systems. *Applied Artificial Intelligence an International Journal*, 9(6):587–621, 1995.

[LWC+16] D. Liu, L. Wang, Y. Chen, M. Elkashlan, K. K. Wong, R. Schober, and L. Hanzo. User Association in 5G networks: A Survey and an Outlook. *IEEE Communications Surveys & Tutorials*, 18(2):1018–1044, 2016.

[MAB+08] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, March 2008.

[mcn17] EU MCN Project; Mobile Cloud Networking. Project reference: 318109. Funded under: FP7. http://www.mobile-cloud-networking.eu/site/, April 2017.

[MF02] N. N. Morsi and A. A. Fahmy. On Generalized Modus Ponens with Multiple Rules and a Residuated Implication. *Fuzzy Sets and Systems*, 129(2):267–274, 2002.

[MFHW12] C. Monsanto, N. Foster, R. Harrison, and D. Walker. A Compiler and Run-time System for Network Programming Languages. In *ACM SIGPLAN NOTICES*, volume 47, pages 217–230, January 2012.

[MGB+09] L. Meyerovich, A. Guha, J. Baskin, G. Cooper, M. Greenberg, and A. Bromfield. Flapjax: A Programming Language for Ajax Applications. In *Proceedings of the 24th ACM conference on Object Oriented Programming Systems Languages and Applications*, volume 44, pages 1–20, New York, NY, USA, October 2009.

[MLK14] W. Meng, W. Li, and L. F. Kwok. EFM: Enhancing the Performance of Signature-based Network Intrusion Detection Systems using Enhanced Filter Mechanism. *Computers & Security*, 43:189–204, 2014.

[MMRAT16] M. Mas, M. Monserrat, D. Ruiz-Aguilera, and J. Torrens. RU and (U, N)-Implications Satisfying Modus Ponens. *International Journal of Approximate Reasoning*, 73:123–137, 2016.

[MMZ+16] C. C. Marquezan, K. Mahmood, A. Zafeiropoulos, R. Krishna, X. Huang, X. An, D. Corujo, F. Leitão, M. L. Rosas, and H. Einsiedler. Context Awareness in Next Generation of Mobile Core Networks. *arXiv preprint arXiv:1611.05353*, 2016.

[Moh15] W. Mohr. The 5G Infrastructure Public-Private Partnership. In *Presentation in ITU GSC-19 Meeting*, 2015.

[MPTSF16] E. Moradi-Pari, A. Tahmasbi-Sarvestani, and Y. P. Fallah. A Hybrid Systems Approach to Modeling Real-time Situation-awareness Component of Networked Crash avoidance systems. *IEEE Systems Journal*, 10(1):169–178, 2016.

[MSG+16] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba. Network Function Virtualization: State-of-the-art and Research Challenges. *IEEE Communications Surveys & Tutorials*, 18(1):236–262, September 2016.

[MVTG14] J. Medved, R. Varga, A. Tkacik, and K. Gray. Opendaylight: Towards a Model-driven Sdn Controller Architecture. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*, 2014.

[NCC+16] P. Neves, R. Calé, M. R. Costa, C. Parada, B. Parreira, J. Alcaraz-Calero, Q. Wang, J. Nightingale, E. Chirivella-Perez, and W. Jiang. The SELFNET Approach for Autonomic Management in an NFV/SDN Networking Paradigm. *International Journal of Distributed Sensor Networks*, 2016:2, 2016.

[NEC15] NEC Corporation. Network Evolution Toward 2020 and Beyond. http://www.nec.com/en/global/solutions/nsp/5g_vision/doc/2020_network.pdf, 2015.

[Net14] E. NetWorld2020. 5G: Challenges, Research Priorities, and Recommendations. *Joint White Paper September*, 2014.

[NLZ14] M. Naderpour, J. Lu, and G. Zhang. A Situation Risk Awareness Approach for Process Systems Safety. *Safety Science*, 64:173–189, 2014.

[NNL15] M. Naderpour, S. Nazir, and J. Lu. The Role of Situation Awareness in Accidents of Large-scale Technological Systems. *Process Safety and Environmental Protection*, 97:13–24, 2015.

[Nok14] Nokia. 5G Use Cases and Requirements White Paper, July 2014.

[nor17] EU 5G-NORMA Project; 5G Novel Radio Multiservice Adaptive Network Architecture. Project reference: 671584 . Funded under: H2020-ICT-2014-2. https://5gnorma.5g-ppp.eu/, April 2017.

[NRFC09] A. Nayak, A. Reimers, N. Feamster, and R. Clark. Resonance: Dynamic Access Control for Enterprise Networks. In *Proceedings of the 1st ACM Workshop on Research on Enterprise Networking*, pages 11–18, New York, NY, USA, August 2009.

[NZGH16] S. Ni, Y. Zhuang, J. Gu, and Y. Huo. A Formal Model and Risk Assessment Method for Security-critical Real-time Embedded Systems. *Computers & Security*, 58:199–215, 2016.

[OBB+14] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, and H. Taoka. Scenarios for 5G Mobile and Wireless Communications: The Vision of the METIS Project. *IEEE Communications Magazine*, 52(5):26–35, 2014.

[oIRT16] Forum of Incident Response and Security Teams. CVSS: Common Vulnerability Scoring System. 2016.

[ONF17] ONF. Open Networking Foundation. https://www.opennetworking.org/, April 2017.

[ope09] OpenFlow Switch Specification v.1.0.0, December 2009.

[ope13] OpenFlow Management and Configuration Protocol (OF-Config) v.1.1.1, March 2013.

[ope17a] OpenBaton. http://openbaton.github.io/, April 2017.

[ope17b] OpenStack. https://www.openstack.org/, April 2017.

[oST07] National Institute of Standards and Technology. NIST-SP800 Series Special Publications on Computer Security. 2007.

[PJ12] S. Paul and R. Jain. Openadn: Mobile Apps on Global Clouds using Openflow and Software Defined Networking. In *2012 IEEE Globecom Workshops*, pages 719–723, 2012.

[Pla17] PlanetLab. https://www.planet-lab.org/, April 2017.

[PNC+14] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, and A. Neal. Mobile-edge computing introductory technical white paper. *White Paper, Mobile-edge Computing (MEC) Industry Initiative*, September 2014.

[POK13] R. Parvizi, F. Oghbaei, and S. R. Khayami. Using COBIT and ITIL Frameworks to establish the Alignment of Business and IT Organizations as one of the Critical Success Factors in ERP Implementation. In *Information and Knowledge Technology (IKT), 2013 5th Conference on*, pages 274–278, 2013.

[PPA+09] B. Pfaff, J. Pettit, K. Amidon, M. Casado, T. Koponen, and S. Shenker. Extending Networking into the Virtualization Layer. In *Proceedings of the ACM SIGCOMM HotNets*, pages 1–6, New York, NY, USA, October 2009.

[PSS16] N. Panwar, S. Sharma, and A. K. Singh. A survey on 5g: The next generation of mobile communication. *Physical Communication*, 18:64–84, 2016.

[PWH13] K. Pentikousis, Y. Wang, and W. Hu. MobileFlow: Toward Software- Defined Mobile Networks. In *IEEE Communications Magazine*, volume 51, pages 44–53, July 2013.

[RFR+12] M. Reitblatt, N. Foster, J. Rexford, C. Schlesinger, and D. Walker. Abstractions for Network Update. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, volume 42, pages 323–334, New York, NY, USA, October 2012.

[RG12] M. Radenkovic and A. Grundy. Efficient and Adaptive Congestion Control for Heterogeneous Delay-tolerant Networks. *Ad Hoc Networks*, 10(7):1322–1345, 2012.

[RMTF09] A. Ramachandran, Y. Mundada, M. B. Tariq, and N. Feamster. Securing Enterprise Networks Using Traffic Tainting, August 2009.

[SA117] 5G-Ensure Project. Enablers for Network and System Security and Resilience. Project reference: 671562 Funded under: H2020-ICT-2014-2. http://www.5gensure.eu/, April 2017.

[SC15] D. Samsung Center. Samsung 5G VisionElectronics Co. 2015.

[sel17] EU SELFNET Project - Self-Organized Network Management in Virtualized and Software Defined Networks. Project reference: H2020-ICT-2014-2/671672. Funded under: H2020. http://www.selfnet-5g.eu, April 2017.

[SKIW17] U. Sivarajah, M. M. Kamal, Z. Irani, and V. Weerakkody. Critical Analysis of Big Data Challenges and Analytical Methods. *Journal of Business Research*, 70:263–286, 2017.

[SKW+15] S. Shanbhag, A. R. Kandoor, C. Wang, R. Mettu, and T. Wolf. VHub: Single-stage Virtual Network Mapping through Hub Location. *Computer Networks*, 77:169–180, 2015.

[SM10] R. Schmittling and A. Munns. Performing a Security Risk Assessment. *ISACA Journal*, 1:10–18, February 2010.

[SNC+14] B. Sonkoly, F. Németh, L. Csikor, L. Gulyás, and A. Gulyás. SDN Based Testbeds for Evaluating and Promoting Multipath TCP. In *Proceedings of the IEEE International Conference on Communications 2014*, pages 3044–3050, Sydney, NSW, Australia, August 2014.

[SRA+16] J. P. Santos, A. Rui, L. Andrade, A. L. Valdivieso Caraguay, L. I. Barona López, and L. J. García Villalba. SELFNET Framework Self-healing Capabilities for 5G Mobile Networks. *Transactions on Emerging Telecommunications Technologies*, 27(9):1225–1232, 2016.

[SSABC16] A. Shameli-Sendi, R. Aghababaei-Barzegar, and M. Cheriet. Taxonomy of Information Security Risk Assessment (ISRA). *Computers & Security*, 57:14–30, 2016.

[SSC+12] S. Sharma, D. Staessens, D. Colle, M. Pickavet, and P. Demeester. A Demonstration of Fast Failure Recovery in Software Defined Networking. In *Testbeds and Research Infrastructure. Development of Networks and Communities*, volume 44, pages 411–414. Thessanoliki, Greece, June 2012.

[SSH15] J. Shin, H. Son, and G. Heo. Development of a Cyber Security Risk Model using Bayesian Networks. *Reliability Engineering & System Safety*, 134:208–217, 2015.

[SSHC+13] S. Sezer, S. Scott-Hayward, P. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao. Are We Ready for SDN? Implementation Challenges for Software-Defined Networks. *IEEE Communications Magazine*, 51(7):36–43, July 2013.

[SV15] H. Seppänen and K. Virrantaus. Shared Situational Awareness and Information Quality in Disaster Management. *Safety Science*, 77:112–122, 2015.

[TC17] A. Talon and C. Curt. Selection of Appropriate Defuzzification Methods: Application to the Assessment of Dam Performance. *Expert Systems with Applications*, 70:160–174, 2017.

[TFF+13] Patricia Thaler, Norman Finn, Don Fedyk, Glenn Parsons, and Eric Gray. Ieee 802.1 q, 2013.

[tno17] EU T-NOVA Project; Network Functions as-a-Service over Virtualised Infrastructures. Project reference: 619520. Funded under: FP7. http://www.t-nova.eu/, April 2017.

[TPL+16] H. Tullberg, P. Popovski, Z. Li, M. Uusitalo, A. Höglund, O. Bulakci, M. Fallgren, and J. F. Monserrat. The METIS 5G System Concept–Meeting the 5G Requirements. *IEEE Communications Magazine*, 54(12):132–139, 2016.

[TSK+17] H. Tahaei, R. Salleh, S. Khan, R. Izard, K. K. R. Choo, and N. B. Anuar. A Multi-objective Software Defined Network Traffic Measurement. *Measurement*, 95:317–327, 2017.

[uni17] EU UNIFY Project; Unifying Cloud and Carrier Networks. Project reference: 619609. Funded under: FP7. https://www.fp7-unify.eu/, April 2017.

[VE16] R. Venkatesan and M. J. Er. A Novel Progressive Learning Technique for Multi-class Classification. *Neurocomputing*, 207:310–321, 2016.

[VKF12] A. Voellmy, H. Kim, and N. Feamster. Procera: A Language for High- Level Reactive Network Control. In *Proceedings of the First Workshop on Hot topics in Software Defined Networks*, pages 43–48, Helsinki, Finland, August 2012.

[vS13] K. van Surksum. Paper: VMware NSX Network Virtualization Design Guide. 2013.

[VW12] A. Voellmy and J. Wang. Scalable Software Defined Network Controllers. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, volume 42, pages 289–290, New York, NY, USA, August 2012.

[WAMS14] J. Webb, A. Ahmad, S. Maynard, and G. Shanks. A Situation Awareness Model for Information Security Risk Management. *Computers & Security*, 44:1–15, 2014.

[WH92] Y.-W. Wang and E.N. Hanson. A Performance Comparison of the Rete and TREAT Algorithms for Testing Database Rule Conditions. In *Proceedings of the Eighth International Conference on Data Engineering*, pages 88–97, August 1992.

[WT01] T. Wolf and J. Turner. Design Issues for High-performance Active Routers. *IEEE Journal on Selected Areas in Communications*, 19(3):404–409, March 2001.

[XAY+16] L. Xu, H. Assem, I. G. B. Yahia, T. S. Buda, A. Martin, D. Gallico, M. Biancani, A. Pastor, P. Aranda, and M. Smirnov. CogNet: A Network Management Architecture Featuring Cognitive Capabilities. In *Proccedings of the European Conference on Networks and Communications*, pages 325–329, 2016.

[YDAG04] L. Yang, R. Dantu, T. Anderson, and R. Gopal. Forwarding and Control Element Separation (ForCES) Framework. RFC 3746 (Informational), April 2004.

[YKLA15] M. Yang, F. Khan, L. Lye, and P. Amyotte. Risk Assessment of Rare Events. *Process Safety and Environmental Protection*, 98:102–108, 2015.

[ZCB10] Q. Zhang, L. Cheng, and R. Boutaba. Cloud Computing: State-of-the-art and Research Challenges. *Journal of internet services and applications*, 1(1):7–18, April 2010.

[zet16] The Zettabyte EraTrends and Analysis, June 2016.

[ZWH+16] T. Zhang, J. Wang, J. Huang, Y. Huang, J. Chen, and Y. Pan. Adaptive Marking Threshold Method for Delay-sensitive TCP in Data Center Network. *Journal of Network and Computer Applications*, 61:222–234, 2016.

Part III

Descripción de la Investigación

Capítulo 10

Introducción

La rápida proliferación del uso de dispositivos móviles ha revelado la incapacidad de las redes actuales para dar soporte a la inmensa cantidad de información que tendrán que gestionar. Esta situación ha dado lugar al desarrollo de una nueva generación de redes móviles que no solo deberá ser capaz de brindar soluciones a dichos problemas, sino también de mejorar las características de sus predecesoras. Capacidades relacionadas con la transferencia masiva de datos, baja latencia, interoperabilidad o la reducción en el consumo de energía, permiten una mejor calidad de experiencia al usuario final. Para alcanzar estas metas se requiere la innovación en diferentes campos, como por ejemplo la provisión de altas tasas de transmisión, una mejor gestión de la información o la introducción de métodos de análisis de datos. Esta última tiene un impacto en los modelos de negocios basados en servicios y en aplicaciones en tiempo real (salud, seguridad, voz sobre IP, transmisión de video, etc). Sin embargo, actualmente el desarrollo de estos servicios está limitado por procesos lentos de estandarización, por el pobre rendimiento en la gestión de la red y en las estrategias aplicadas en la toma de cisiones. Para cubrir dichas necesidades, la quinta generación de redes móviles, o 5G, propone la combinación de algunas tecnologías emergentes tales como Redes Definidas por Software (del inglés SDN), Virtualización de la Funciones de Red (Del inglés NFV), computación en la nube, Redes auto-organizadas (del inglés SON), aprendizaje automático, inteligencia artificial, entre otras. SDN se basa en la separación del plano de control del plano de datos en los dispositivos tradicionales de red. Esta descomposición permite el control centralizado de la red, proporcionando capacidades de automatización y la simplificación de las tareas de gestión de red, mientras acelera la innovación de nuevas aplicaciones de alto nivel. Por su parte, NFV permite la implementación de las tradicionales Funciones de Red (del inglés NF) como instancias virtuales, las cuales se ejecutan sobre hardware genérico. El enfoque de escalabilidad provisto por NFV permite que las Funciones de Red Virtuales (del inglés VNF) puedan ser instanciadas en cualquier momento y lugar en un tiempo mucho menor si se compara con el proceso tradicional de despliegue de funciones de red. Desde el punto de vista técnico, SDN y NFV son tecnologías complementarias, que en conjunto facilitan la configuración y la personalización de los servicios de red. Asimismo, conceptos como computación en la nube, Inteligencia Artificial y SON facilitan el despliegue de servicios (bajo demanda) y permiten una mejor gestión del tráfico de la red basado en decisiones

inteligentes. Por otra parte, existe la tendencia de introducir metodologías cognitivas para facilitar la comprensión del sistema a través del análisis contextual de la información tal es el caso del modelo de Conciencia Situacional (del inglés SA) propuesto por Endsley. De acuerdo con éste modelo, la percepción, comprensión y proyección del estado del sistema deben ser tomados en cuenta para conocer que está sucediendo realmente en el entorno protegido y como evitar o mitigar posibles problemas. En este contexto, esta investigación introduce un Marco de Análisis capaz de diagnosticar diferentes problemas en redes 5G. Esta propuesta es consiente del contexto situacional de los elementos y las aplicaciones relacionadas a redes 5G, y en consecuencia es capaz de proveer capacidades de inteligencia. Las siguientes secciones resumen el problema de investigación y los objetivos de esta tesis. De igual forma, las principales contribuciones son presentadas.

10.1 Problema de Investigación

El crecimiento exponencial de los servicios en línea (transmisión de video, banca en línea, etc) y del número de dispositivos conectados ha generado nuevos desafíos para la gestión de la infraestructura actual de red en términos de seguridad, rendimiento y confiabilidad. La gestión y respuesta rápida frente a problemas de red inesperados (caída de enlaces, congestión, ataques de DDoS, retardo) son fundamentales para garantizar Calidad de Servicio (del inglés QoS) y Calidad de Experiencia (del inglés QoE) a los usuarios finales, mientras el tiempo de recuperación de servicios y el costo de capital y de operación disminuyen (capex and opex). En la actualidad la personalización de los servicios de red requiere la configuración individual de cada dispositivo. Por su parte, la introducción de soluciones nuevas está limitada por la rigidez de las arquitecturas tradicionales debido a que el proceso de estandarización toma un tiempo considerable (desde su diseño hasta su implementación). De igual forma, los mecanismos de análisis de datos y de inteligencia de red son fundamentales para resolver o mitigar problemas potenciales. Debido a esto, la provisión de capacidades de análisis de datos e inteligencia en redes 5G son temas clave para la comunidad de investigación, siendo ésta la principal motivación del presente trabajo. Actualmente existen diferentes proyectos e iniciativas que pretenden cubrir los requerimientos de inteligencia y auto gestión en este tipo de escenarios. Es importante destacar que los primeros avances en 5G se esperan a partir del 2020 y por tanto son trabajos aún en progreso.

10.2 Objetivos

Teniendo en cuenta las necesidades de dinamismo e inteligencia de las redes 5G, el principal objetivo de la presente investigación consiste en la provisión de un modelo de conciencia situacional para el análisis de datos en ambientes 5G. Esta propuesta presta atención especial al análisis de datos, la predicción, el reconocimiento de patrones, la aplicación de umbrales adaptativos y capacidades para la inferencia de conocimiento. La idea fundamental detrás de estas acciones es facilitar el proceso de toma de decisiones para la resolución y mitigación de problemas comunes de red tanto de forma reactiva

como proactiva. Tomando en cuenta la necesidad de conocer el contexto de la situación en una red 5G, se han cubierto los siguientes objetivos durante esta investigación:

1. En primer lugar, se revisa el estado del arte relacionado con SDN, NFV y los requerimientos de 5G para definir qué elementos deben ser tomados en cuenta en el proceso de diagnóstico y cómo las tecnologías actuales permiten cubrir dichas necesidades. Los requerimientos y las ventajas de incorporar capacidades de análisis de datos son también discutidos.

2. Como segundo paso, se describe de manera general el proyecto SELFNET, lo cual permite identificar las fuentes de información a ser considerados en el proceso de análisis. Este apartado incluye las entradas/salidas y los diferentes requerimientos del Marco de Análisis de SELFNET.

3. Luego, el modelo de conciencia situacional para redes 5G es propuesto. Este enfoque toma en cuenta las necesidades de inteligencia y provee una perspectiva completa del estado de la red, facilitando la gestión de incidencias y el proceso de toma de decisiones.

4. Una vez que el contexto situacional y los requerimientos para el análisis de datos son definidos, se propone el Marco de Análisis de SELFNET. Este marco propone una solución escalable y modular orientado a casos de uso, El Marco de Análisis de SELFNET identifica situaciones sospechosas o inesperadas basándose en las métricas proporcionadas por los diferentes componentes de una red 5G, las reglas de análisis y otros parámetros definidos por el operador del caso de uso.

10.3 Resumen de las Contribuciones de la Tesis

Los resultados del presente trabajo son organizados en cuatro áreas de conocimiento: i) Redes Definidas por Software (SDN) y Virtualización de las Funciones de Red (NFV), ii) Redes Móviles 5G, iii) Conciencia Situacional y Gestión de Incidencias y IV) Análisis de Datos en redes 5G. Las contribuciones de la presente tesis se muestran en el esquema de la Figura 10.1.

Teniendo en cuenta el objetivo principal de este trabajo, que es la provisión de un marco de Análisis consiente del contexto situacional de una red 5G, cuatro campos son estudiados. Esta tesis permite el análisis de datos y la auto-gestión de redes 5G en base a tecnologías de red emergentes, tal es caso de SDN y NFV. En este sentido, las contribuciones enfocadas en SDN y NFV son presentadas en [BLVCGV15] y [BLVCGVL15]. Luego, la contribución [BLVCSMGV16] presenta la aplicación de SDN y NFV en ambientes 5G y muestra el estado actual alrededor de esta línea de investigación. Tomando en cuenta las contribuciones mencionadas anteriormente, esta tesis propone un enfoque general para ayudar en el proceso de gestión de incidencias a través de las contribuciones [BLMVVC+16] y [BLVCMV+17]. A su vez, este enfoque define una etapa de análisis mediante el cual se conoce el estado real de la red. Para cumplir dicho propósito, el Marco de Análisis de SELFNET y sus componentes son presentados en las contribuciones [BLMVGV17a] y

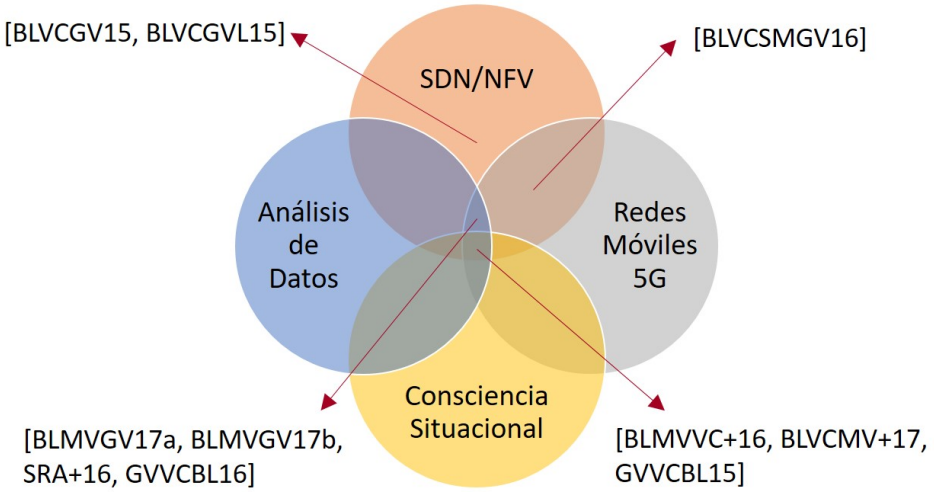

Figura 10.1: Contribuciones de esta tesis.

[BLMVGV17b]. De acuerdo al estado de arte relacionado a la temática, El Marco de Análisis de SELFNET es la primera propuesta que integra ambos conceptos el análisis de datos y el contexto situacional en redes 5G. Estas iniciativas se encuentran en línea con las capacidades del proyecto SELFNET [SRA+16], [GVVCBL15] y [GVVCBL16].

10.4 Estructura del Trabajo

Esta tesis es organizada como sigue:

En el capítulo 10 se muestra la información general de este trabajo, el problema de investigación, los objetivos y las contribuciones de la tesis.

El capítulo 11 describe el proyecto "Self-organized Network Management in Virtualized and Software Defined Networks (SELFNET)". El capítulo muestra la arquitectura de SELFNET y sus componentes, con lo cual se establece las bases de un Marco de auto gestión para infraestructuras SDN/NFV.

El capítulo 12 presenta una arquitectura para la gestión de incidencias en redes 5G. Además, describe la aplicabilidad del Modelo de Conciencia Situacional de Endsley en esquemas de gestión de riesgos tradicionales, tomando en cuenta las diferentes fuentes de información de las redes 5G.

En el capítulo 13 se detalla el Módulo de Análisis de SELFNET. Se describe como el enfoque propuesto facilita el proceso de toma de decisiones. Con este propósito, se describen los principios de diseño, la especificación de datos, la arquitectura de alto nivel y sus componentes. Finalmente se introduce la utilización de técnicas avanzadas de análisis tales como predicción, reconocimiento de patrones o la inferencia de conocimiento.

Finalmente, el capítulo 14 presenta las conclusiones y el trabajo futuro derivado de esta tesis.

10.5 Audiencia de la Tesis

El estado del arte proporciona información y definiciones relacionadas a redes 5G, lo cual permite al lector entender la contribución principal de esta investigación. Los requisitos previos para entender adecuadamente el material de esta tesis no son elevados. Un conocimiento básico sobre sistemas móviles, protocolos, gestión de incidencias y análisis de datos es requerido. Este trabajo presenta un modelo de conciencia situacional para el análisis de datos en redes 5G junto con una extensa bibliografía, la cual proporciona información adicional que puede ser consultada en lo posterior por el lector.

Capítulo 11

SELFNET Gestión Autónoma en Redes SDN/NFV

Este capítulo hace una revisión de los principales avances en redes autogestionadas basadas en SDN/NFV. Asimismo, el proyecto SELFNET y sus componentes son descritos en los próximos apartados. Este capítulo está organizado en 10 secciones. La sección 11.1 hace una introducción al capítulo. La sección 11.2 hace una revisión de la gestión de redes con SDNNFV. En la sección 11.3 presenta el proyecto SELFNET. Las secciones siguientes describen cada una de las capas y subcapas del framework SELFNET. La sección 11.4 describe la Capa de Infraestructura. En la sección 11.5 se describe la Capa de Datos de Red, mientras que la sección 11.6 presenta la Capa de Control SON. A su vez, la sección 11.7 describe la Capa Autónoma SON. Por otra parte, la sección 11.8 presenta la Capa de Gestión y Orquestación NFV y en la sección 11.9 se describe la Capa de Acceso SON. Finalmente, la sección 11.10 presenta el resumen de este capítulo.

11.1 Introducción

La gestión y personalización de servicios de red se ha visto limitada por la rigidez de las arquitecturas de red tradicionales y por el incremento tanto de los costes de capital como operacionales. En la actualidad, la solución a problemas de red comunes, como fallos en los enlaces, ataques de seguridad, degradación de la QoS o de la QoE, congestión, entre otros, requieren la intervención directa de los operadores de red. La reconfiguración manual de dispositivos o la instalación de nuevos equipos (encaminadores, servidores NAT, cortafuegos) compromenten la operatividad de la red y repercuten negativamente sobre los niveles de servicio acordados en los SLAs. De forma similar, la creación de servicios innovadores de valor añadido se ve limitada por el hardware y software propietarios y que, en algunos casos, deben únicamente pertenecer al mismo proveedor. Esas limitaciones hacen que las arquitecturas de redes tradicionales se muestren inviables para satisfacer las necesidades actuales de los usuarios, las empresas y los operadores de red.

La solución propuesta para hacer frente a los desafíos actuales ha sido conducida por los avances logrados por la ingeniería de software, en la que los desarrolladores pueden crear sus propias aplicaciones usando lenguajes de programación de alto nivel. Estos programas

pueden ser ejecutados en diversos equipos debido a la abstracción de recursos que los OS son capaces de brindar. En este contexto, SDN y NFV se muestran como potenciales alternativas para lidiar con los recientes desafíos. SDN propone el desacoplamiento de los planos de control y datos en los dispositivos de red, posibilitando su desarrollo y evolución independiente, además de una visión centralizada de la red. NFV promueve la migración de funcionalidades de red típicamente desplegadas en dispositivos (DPI, cortafuegos, balanceadores de carga) a paquetes de software o funciones de red NF que puedan ser instanciadas en una infraestructura virtualizada. Ambas arquitecturas son complementarias y potencialmente integrables para ofrecer a los desarrolladores un entorno de red abierto. Este capítulo describe SDN, NFV, y su evolución en los últimos años. Asimismo analiza la gestión de redes, sus oportunidades y desafíos en el futuro.

Adicionalmente, tanto el crecimiento exponencial de los dispositivos móviles como el advenimiento de la computación en la nube trajeron consigo desafíos adicionales a los operadores de red y proveedores de servicio. Se requiere también, un decremento radical de operaciones de gestión de red integradas que no afecten negativamente la calidad de servicio QoS/QoE ni la seguridad. De forma similar, se promueve un nuevo modelo que integre el acceso y la gestión de recursos móviles. Se espera que las futuras redes 5G proporcionen no sólo una mejor ancho de banda, sino también un modelo de control heterogéneo, simplificado y unificado. Los costes de gestión deben ser reducidos a través de la automatización de operaciones. En este contexto, se presenta como desafío fundamental la reducción de costos operacionales por medio del desarrollo de arquitecturas de gestión escalables que incluyan técnicas de minería de datos, reconocimiento de patrones, algoritmos de aprendizaje automático, etc. Este capítulo describe también el estado del arte y los recientes avances en este campo.

El proyecto SELFNET [sel17] hace uso de los principios de SDN y NFV para proporcionar una arquitectura de gestión autónoma de funciones de red. De este modo, se facilita la resolución de problemas de red y se mejora la calidad de servicio QoS percibida por los usarios. La auto-gestión es facilitada por medio del uso minería de datos, algoritmos de aprendizaje automático, reconocimiento de patrones, etc. acoplados a entornos móviles 5G. Asimismo, el sistema es capaz de decidir las mejores acciones que mitiguen de forma automática problemas de red. La arquitectura de SELFNET está compuesta por diversas capas: Infraestructura, Red Virtualizada, Control SON, Acceso a la Red y SON Autónoma. Dentro de la capa SON Autónoma, la subcapa de Monitorización y Análisis es una de las que presenta importantes desafíos. Dicha subcapa, a su vez, está dividida en tres módulos: Monitorización y Descubrimiento, Aggregación y Correlación, y Análisis.

11.2 Gestión en Redes SDN/NFV

Los principios de SDN y NFV ofrecen diversas ventajas sobre as arquitecturas de gestión de red tradicionales. Diversos consorcios conformados por operadores de red, universidades, centros de investigación, proveedores de servicios, entre otros, han concentrado sus esfuerzos en el desarrollo de arquitecturas de gestión innovadoras sobre entornos de red virtualizados. En la Tabla 11.1 se describen proyectos relevantes de gestión basados en

SDN y NFV.

Cuadro 11.1: Proyectos para la gestión de red basados en SDN/NFV

Proyecto	Dominio	Descripción	Escenario de Aplicación
CROWD [cro17]	SDN, SON	Este proyecto tiene el objetivo de aumentar la capacidad en densidad de las redes de acceso inalámbrico heterogéneas. Asimismo, se centra en garantizar la QoE de los usuarios móviles, la optimización de los mecanismos MAC y el consumo de energía. De esta forma, se mejora la gestión del tráfico en redes inalámbricas con alta densidad.	Gestión de Tráfico
5G-NORMA [nor17]	SDN, NFV	Este proyecto se centra en proporcionar la capacidad de adaptación de un recurso de manera eficiente. El framework gestiona las fluctuaciones en la demanda de tráfico por medio de un portafolio de aplicaciones de servicios. Las nuevas funciones de red ofrecen el soporte eficiente de recursos en distintos escenarios, y ayudan a incrementar la eficiencia energética.	Escenario Multi-servicio, Escenario Multi-tenant
MCN [mcn17]	SDN	El proyecto se centra en la mejora del procesamiento del tráfico mediante la separación entre hardware de radio y hardware de reenvío de paquetes.	Entorno SDN
UNIFY [uni17]	SDN, NFV	El proyecto tiene como objetivo desarrollar una plataforma de creación automatizada y dinámica de servicios a través de la creación de un modelo de servicios y un lenguaje de relación de servicios. El proyecto permitirá el despliegue dinámico y automático de los servicios en recursos de red, computación y de almacenamiento que se encuentran disponibles en la infraestructura. De manera similar, el orquestador incluirá algoritmos de optimización para asegurar la colocación óptima de componentes de servicio elementales a lo largo de la infraestructura.	Infraestructura. Virtualización Encadenamiento flexible de servicios. Invocación de cadena de servicios de red para proveedores
T-NOVA [tno17]	SDN, NFV	Este proyecto se centra en el despliegue de Funciones de Red como Servicio (NFaaS) sobre infraestructuras de red virtualizadas. Para este propósito, se diseña e implementa una plataforma de gestión y orquestación para la provisión automatizada, configuración, monitorización y optimización de recursos virtualizados. Además, SDN se utiliza para la gestión eficiente de la infraestructura de red.	Escenario de Alto Nivel, Escenario de concatenación de NFV.

11.3 Arquitectura SELFNET de Gestión Autónoma para Redes SDN/NFV

El proyecto SELFNET perteneciente al programa H2020 tiene el propósito de diseñar e implementar un framework de gestión autónomo que provea capacidades auto-organizativas SON en las nuevas infraestructuras móviles 5G. A través de la detección y mitigación automática de problemas comunes de red, que actualmente son tratados de forma manual por los administradores de red, SELFNET proveerá un framework capaz de reducir de forma significativa los costes operacionales y, en consecuencia, mejorar la experiencia de usuario [sel17], [NCC+16].

Mediante la integración de tecnologías innovadoras como SDN, NFV, SON, Cloud Computing, Inteligencia Artificial, QoS/QoE y conceptos de redes de nueva generación; SELFNET proveerá un sistema de gestión de red inteligente, escalable y extensible. Este framework asistirá a los operadores de red en el desempeño de tareas de gestión, tales como el despliegue automático de aplicaciones SDN/NFV que provean capacidades de

monitorización y mantenimiento autónomo de la red. Directivas y políticas de gestión de alto nivel pondrán en marcha acciones que mitiguen problemas actuales o potenciales amenazas futuras. SELFNET abordará tres principales escenarios de gestión de red: la provisión de capacidades de auto-protección (self-protection) contra ataques de red distribuidos, capacidades de auto-recuperación (self-healing) contra fallos en la red, y capacidades de auto-optimización (self-optimization) que mejoren dinámicamente el rendimiento de la red y la QoE. Estas funcionalidades provistas por SELFNET otorgarán los fundamentos para cumplir con algunos de los requierimientos de 5G, definidos por el consorcio 5G-PPP.

En este contexto, la Figura 11.1 ilustra la arquitectura SELFNET. Esta arquitectura está basada en seis capas diferenciadas con los siguientes alcances a nivel lógico: Capa de Infraestructura, Capa de Datos de Red, Capa de Control SON, Capa Autónoma SON, Capa de Gestión y Orquestación NFV y Capa de Acceso SON. En las siguientes secciones, cada capa es descrita.

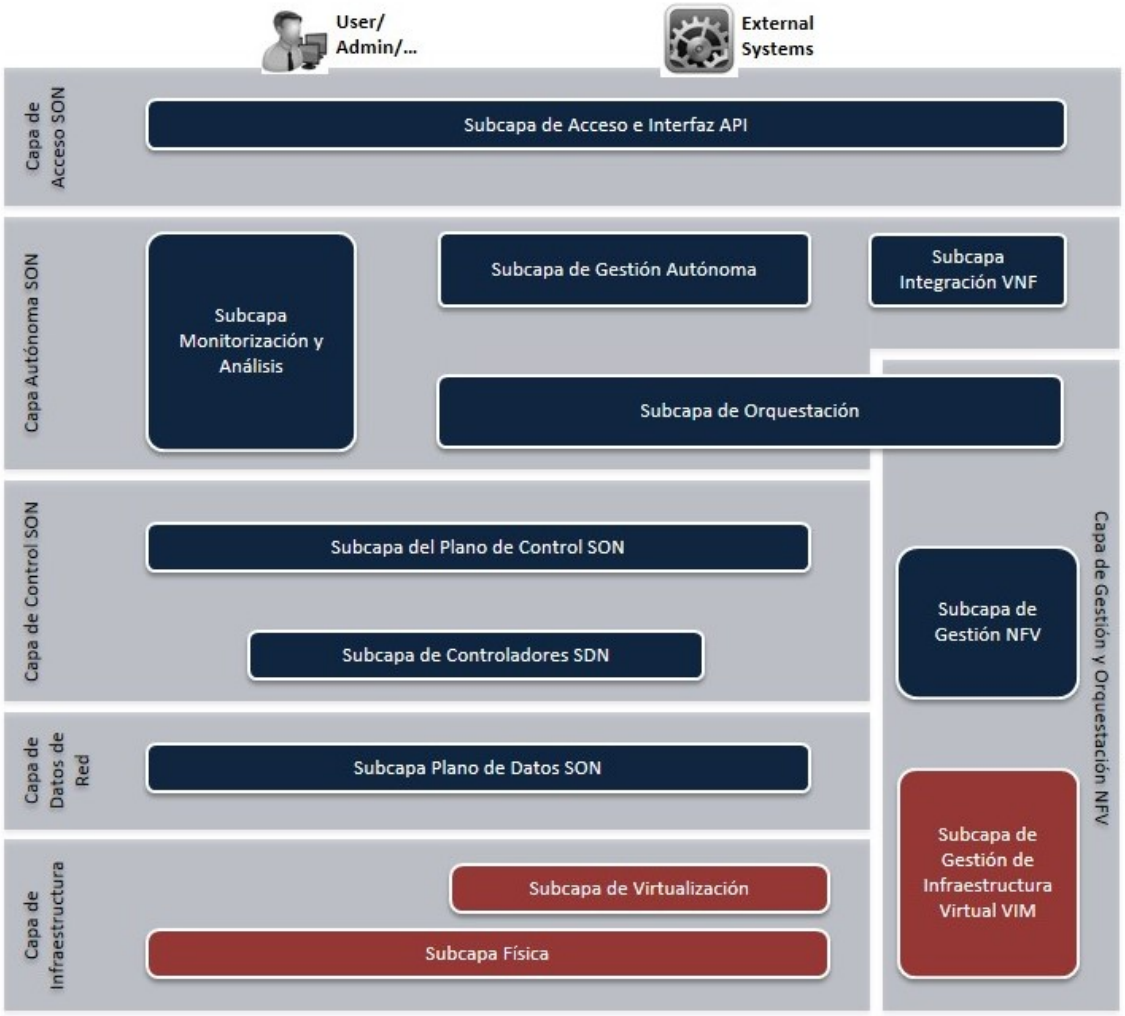

Figura 11.1: Vista general de la arquitectura SELFNET [NCC+16]

11.4 Capa de Infraestructura

Esta capa provee los recursos necesarios par la instanciación de funciones virtuales (computacionales, red, almacenamiento) y soporta los mecanismos necesarios para hacerlo. Aquello representa el componente NFVI definido en la terminología ETSI NFV [ETS13a]. Para lograr su operatividad, se definen dos subcapas: Subcapa Física y Subcapa de Virtualización.

11.4.1 Subcapa Física

La Subcapa Física incluye los recursos físicos requeridos para proveer capacidades computacionales, de red y de almacenamiento sobre hardware bare metal. Debido a que SELFNET está diseñado para operar sobre 5G, los elementos físicos siguen una arquitectura de borde (mobile edge) en la que los operadores pueden desplegar servicios operacionales y de gestión. El modelo MEC propuesto por ETSI [PNC+14] se ilustra en la Figura 11.2. Dicho modelo propone que los nodos de borde (edge) se encuentren geográficamente separados del centro de datos. De este modo, ciertos servicios pueden ser desplegados tanto cerca del usuario, así como en el centro de datos en caso de requerir alto rendimiento. Adicionalmente, la integración de despliegues de borde (como C-RAN) dentro de MEC quiebra la rigidez típica y facilita la personalización de servicios. Asimismo, se considera que la conectividad entre sus elementos permite capacidades de virtualización alineadas con los avances de 5G.

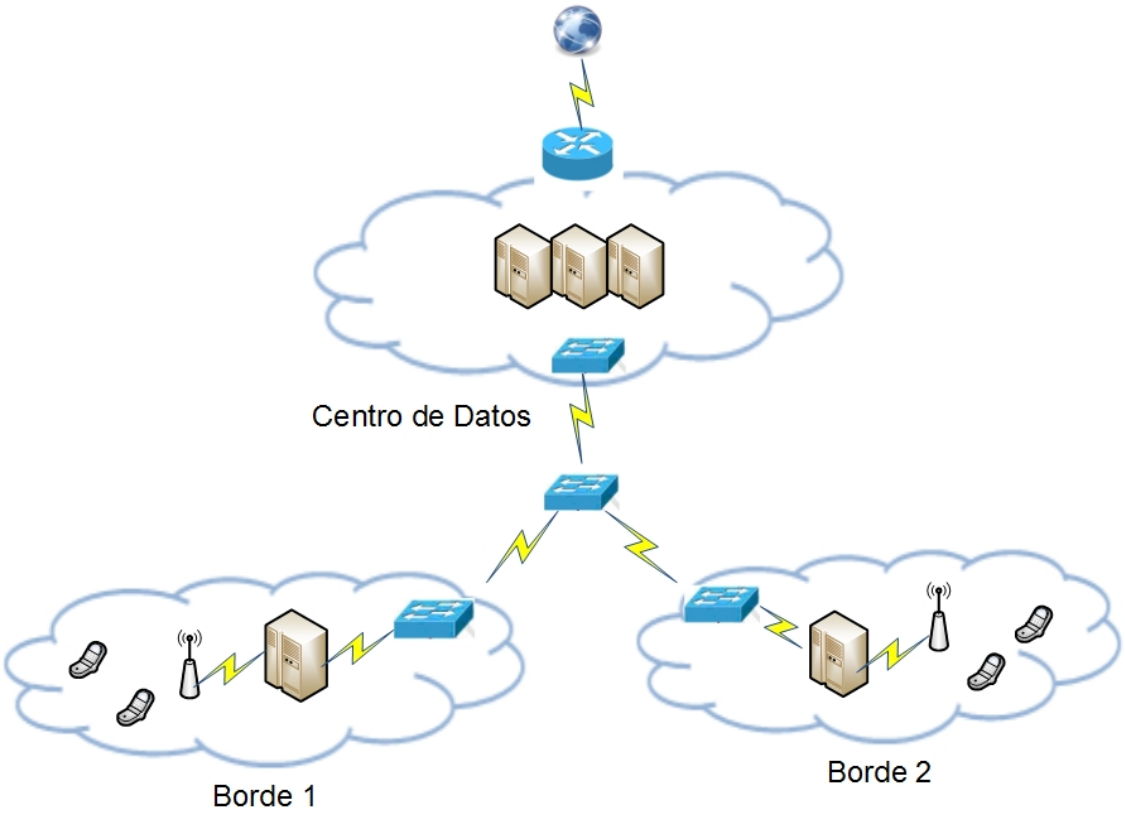

Figura 11.2: Capa física de una infraestructura MEC para redes 5G

11.4.2 Subcapa de Virtualización

La Subcapa de Virtualización posibilita la compartición de los recursos disponibles entre distintos usuarios o servicios. De este modo, se otorga diversas ventajas, tales como el aislamiento, confiabilidad, adaptabilidad y control de los recursos. Sin embargo, la principal desventaja incluye la penalización en el rendimiento producto de las tareas de virtualización. En este contexto, avances recientes en el ámbito de la virtualización cumplen con las expectativas de las infraestructuras 5G con respecto al rendimiento de los recursos virtualizados en entradas y salidas (I/O) [AGHE+15]. En otras palabras, la penalización en el rendimiento por el uso de la virtualización puede considerarse despreciable para los dispositivos modernos. En SELFNET, las capas de virtualización incluyen el uso de switches virtuales para la conexión de máquinas virtuales desplegadas sobre recursos físicos.

11.5 Capa de Datos de Red

En esta capa, las distintas funciones de red son situadas e interconectadas sobre una determinada topología. Las funciones de red NFs incluyen las instancias requeridas para la normal operatividad de la infraestructura virtual así como aquellas creadas por SELFNET como parte de las funcionalidades SON. Debido a que los bordes (edges) y centro de datos están completamente virtualizados, las NFs pueden ser dinámicamente asignadas en ambas ubicaciones.

La Capa de Datos de Red también proporciona soporte multi-dominio que posibilita la compartición de recursos entre distintos dominios, cada cual con su propio sistema de gestión según su modelo de negocio. En las arquitecturas 5G, los recursos pueden ser adquiridos por una alianza entre operadores de telecomunicaciones, y posteriormente son compartidos de acuerdo a sus necesidades. En este contexto, un administrador particular no es capaz de gestionar los recursos de otro operador, o interceptar el tráfico provisto por otro operador.

11.6 Capa de Control SON

Esta capa incluye los elementos responsables de la recolección de datos desde distintas fuentes virtualizadas (sensores SON) y las funciones que ejecutan acciones en la red (actuadores SON). Los sensores y actuadores SON son controlados por la Capa Autónoma SON, la cuál otorga autonomía a la red. De forma similar, la Capa de Control SON interactúa con el plano de control SDN. En otras palabras, traduce políticas globales de gestión autónoma en configuraciones específicas para los elementos de la red.

11.6.1 Subcapa de Controladores SDN

La Subcapa de Controladores SDN representa un controlador lógicamente centralizado (plano de control SDN). El cual gestiona los elementos de la red y controla las funciones que se ejecutan en dichos elementos. El controlador SDN utiliza una interfaz para configurar las

reglas que deben ser ejecutadas en los elementos de red. De este modo, el tráfico que circula a través de dichos dispositivos puede ser dinámicamente modificado. Adicionalmente, el controlador puede modificar funciones adicionales de los dispositivos, por ejemplo protocolos como OpenFlow, OFConfig, NETCONF . Los distintos servicios que ofrece un controlador SDN son desplegados a través de Aplicaciones o SDN-Apps. Ejemplos de SDN-Apps incluyen protocolos de encaminamiento, protocolos de etiquetado o filtrado. En la interfaz superior (northbound), el controlador SDN proporciona una API que permite la gestión, configuración y monitorización remota del comportamiento de la red.

11.6.2 Subcapa del Plano de Control SON

La Subcapa del Plano de Control SON instancia las distintas funciones de red NF que se ejecutan en la infraestructura virtualizada. En la arquitectura SELFNET existen dos tipos de NFs: Sensores SON y Actuadores SON.

- **Sensores SON**. Recopilan datos relacionados con las actividades de red. La información recolectada incluye métricas relacionadas con el tráfico global (por ejemplo, estado de los enlaces, ancho de banda) o métricas específicas (por ejm. DPI, QoS de una transmisión de video relacionado a un flujo de datos específico). Los operadores y proveedores de servicios pueden desarrollar distintos sensores de acuerdo a sus necesidades.
- **Actuadores SON**. Ejecutan un conjunto de acciones específicas sobre el tráfico que circula en la red. Las acciones dependen de la aplicación desarrollada por el proveedor de servicios. Por ejemplo, si el sistema detecta un ataque DDoS, un actuador SON puede automáticamente bloquear el origen de dicho ataque. A su vez, si el sistema detecta degradación de la calidad de servicio, otro actuador SON puede optimizar los flujos de tráfico incrementando la prioridad o el ancho de banda.

11.7 Capa Autónoma SON

Esta capa es responsable de proporcionar inteligencia a la red. La información recolectada por los sensores es usada para diagnosticar el estado de la red. Luego, las acciones para alcanzar los objetivos del sistema son determinadas y ejecutadas. Los principales componentes de la Capa Autónoma SON se describen a continuación.

11.7.1 Subcapa de Monitorización y Análisis

Esta subcapa recolecta los datos provistos por los sensores, los cuales son agregados y correlacionados para extraer información relevante. El analizador usa dicha información para detectar situaciones sospechosas en la red (detección de una botnet, degradación de QoS/QoE, ataques DDoS, caída de enlaces). El proceso completo se organiza en tres fases: Monitorización y Descubrimiento, Agregación y Correlación, y Análisis.

- **Monitorización y Descubrimiento**. Se encarga de recolectar los datos enviados por los sensores SON. Para este propósito, cuando un nuevo sensor es instanciado,

se recibe una notificación y los detalles de dicha instanciación para establecer una conexión que permita luego recibir las métricas correspondientes. Adicionalmente, se recibe la información provista por las subcapas física y virtual. Luego, la información es almacenada en una base de datos con el fin de facilitar su procesamiento en capas superiores.

- **Agregación y Correlación**. Lleva a cabo la correlación y agregación de la información almacenada en la base de datos de monitorización. Este proceso involucra acciones adicionales, como la normalización de datos, verificación y eliminación de información redundante. Al finalizar esta etapa, sólo la información relevante será procesada por el módulo de Análisis.

- **Análisis**. Su objetivo principal es el análisis exhaustivo de la información relevante proporcionada por la capa de Agregación y Correlación. El análisis incluye la predicción de futuros problemas de red inferidos a partir de métricas HoN. Para este propósito, se aprovechan las ventajas de diversos algoritmos de predicción, reconocimiento de patrones y técnicas de big data. Los valores inferidos permiten la aplicación de acciones preventivas y correctivas en el sistema. En esta etapa, los eventos son enviados a la Subcapa de Gestión Autónoma para establecer las correspondientes acciones en la red.

11.7.2 Integración de VNF

Actúa como un repositorio de diferentes funciones de red (NFs). En esta subcapa, las funciones de red disponibles son almacenadas para que sean desplegadas como parte de una acción preventiva o correctiva. A su vez, los proveedores de servicios pueden diseñar, crear y actualizar sus propias aplicaciones. En este contexto, el encapsulamiento de NFs sigue las recomendaciones del framework ETSI MANO [ETS14]. Consecuentemente, el gestor de NFV (VNFM) es un componente clave para todo el ciclo de vida de los sensores y actuadores SON. El ciclo de vida VNFM expone un conjunto común de primitivas para la instanciación, configuración, reconfiguración y eliminación automática de las distintas VNFs. Una API común facilita a los proveedores el fácil diseño y desarrollo de sus soluciones. Una vez que la solución NF es publicada en el repositorio (onboarding), el Gestor Autónomo utiliza dichas funcionalidades para proveer un nuevo servicio (sensor/actuador).

11.7.3 Subcapa de Gestión Autónoma

Este componente usa distintos algoritmos para diagnosticar la causa de un problema de red sobre la base de las métricas HoN provistas por el Analizador. Una vez que la causa es detectada, el Gestor Autónomo usa las NFs disponibles que ofrece el repositorio de VNFs para decidir la mejor estrategia de reacción, o una contramedida (por ej. El despliegue de un nuevo balanceador de carga, cortafuegos o DPI). Luego, las acciones decididas son comunicadas a la Capa de Gestión y Orquestación NFV. Las tareas relacionadas con la Gestión Autónoma están agrupadas en tres módulos.

- **Diagnosticador**. Este elemento diagnostica la causa de los problemas de red a partir de la información proporcionada por la Subcapa de Monitorización y Análisis (topología, datos de sensores, métricas HoN). Asimismo, aprovecha las ventajas de algoritmos estocásticos, inteligencia artificial y minería de datos para determinar el origen del problema. Finalmente, la causa es notificada al submódulo de Toma de Decisiones.

- **Toma de Decisiones**. Recoge la información procedente del Diagnosticador y decide el conjunto de acciones preventivas o correctivas a ser desplegadas con el fin de mitigar los problemas de red detectados. Del mismo modo, este componente aprovecha también la integración de algoritmos de inteligencia artificial para determinar las respuestas o tácticas a ser desplegadas. Las acciones tomadas son notificadas al Ejecutor de Acciones.

- **Ejecutor de Acciones**. Proporciona un conjunto de acciones consistentes a ser ejecutadas en la infraestructura. En otras palabras, valida, organiza y refina las tácticas para evitar conflictos, duplicidad y orden incoherente de las acciones. Al finalizar esta etapa, una descripción de alto nivel de la localización, tipo de actuador SON y parámetros de configuración asociados son transferidos al Orquestador.

11.8 Capa de Gestión y Orquestación NFV

Esta capa es responsable del control y concatenación de las distintas NFs en la infraestructura virtualizada. La arquitectura sigue las recomendaciones [ETS14] y, consecuentemente, está compuesta de: Orquestación, Gestión de VNFs y Gestión de Infraestructura Virtualizada VIM. Tal como se describe en la sección 11.7.2, las operaciones de Gestión de VNFs son parcialmente desarrolladas en el Onboarding de VNFs. El resto de operaciones se describe a continuación:

- **Subcapa de Orquestación y Gestión NFV**. Es responsable de recibir el conjunto de acciones del Gestor Autónomo y orquestar las correspondientes funciones de red sobre los recursos virtuales disponibles. La coordinación y programación de la ejecución de diferentes acciones se realiza mediante la interacción con el Gestor de Infraestructura Virtual.

- **Subcapa de Gestión de Infraestructura Virtual (VIM)**. Se encarga de organizar y proporcionar los recursos virtuales para la instanciación de las diferentes funciones de red. El VIM interactúa con la infraestructura física y virtual para asegurar la disponibilidad de recursos, y realizar el despliegue automático de servicios.

11.9 Capa de Acceso SON

Esta capa proporciona una interfaz atractiva e intuitiva que proporciona diferentes capacidades de monitorización y operación dependiendo de los niveles de privilegios

de los usuarios. De esta forma, los usuarios pueden comprobar el estado actual de las operaciones en SELFNET. Asimismo, la API de acceso registra los sensores y actuadores SON actualmente desplegados en SELFNET, así como las sesiones iniciadas y mensajes, que permiten una visión más amplia del estado de la red. Esta interfaz es usada por actores externos como Sistemas de Soporte Empresarial (BSS) o Sistemas de Soporte Operacional (OSS).

Como se describió en las secciones anteriores, SELFNET pretende ser una solución independiente y autónoma que ayude en la mitigación o resolución de problemas de red sin intervención de los administradores de red. De este modo, la Capa de Acceso SON proporciona también a los usuarios el estudio de las acciones ejecutadas por SELFNET, posibilitando la validación de acciones correctivas de las aplicaciones.

11.10 Resumen

Este capítulo hace un resumen de los avances en auto-gestión de redes basadas en los principios SDN/NFV. Luego, el proyecto SELFNET es presentado. A continuación, las distintas capas y subcapas del framework SELFNET son descritas. Las capas de Infraestructura, Datos de Red, Control SON, Capa Autónoma SON, Gestión y Orquestación NFV y Capa de Acceso SON son presentadas.

Capítulo 12

Conciencia Situacional en 5G

Este capítulo propone una arquitectura para la gestión de incidencias en redes 5G. La propuesta combina las bases de los esquemas de gestión de riesgos convencionales con el modelo de Consciencia Situacional propuesto por Endsley. Se ha tomado en cuenta diferentes aspectos, tales como la capacidad de adaptación a entornos de monitorización dinámicos, el seguimiento de contramedidas o la repercusión del contexto de la incidencia en el proceso de toma de decisiones. Se cubre también todos los niveles de procesamiento de la información en redes móviles, desde la infraestructura de red hasta el despliegue de los actuadores encargados de aplicar las contramedidas.

El resto de este capítulo está estructurado en 4 secciones, presentándose en la primera de ellas (12.1) la introducción y conceptos generales relacionados con la gestión de riesgos y el modelo de conciencia situacional. En la sección 12.2 se discuten las dificultades en la gestión de incidencias en redes 5G. En la sección 12.3 se propone una arquitectura para la gestión de incidencias para redes 5G. Finalmente en la sección 12.4 se resume el presente capítulo.

12.1 Introducción

5G sienta sus bases en la combinación de tecnologías emergentes tal es el caso de SDN, NFV o inteligencia artificial [PSS16]; las cuales permitirán la personalización de los servicios y la gestión eficiente de los elementos que la componen. Sin embargo, el desarrollo de estos servicios está limitado por la falta de estrategias eficientes de gestión y toma de decisiones [IZAD14], lo que conlleva mayor dificultad a la hora de desplegar medidas para el control de incidencias.

En los entornos de monitorización actuales, la gestión de la seguridad de la información habitualmente se lleva a cabo mediante la aplicación de directivas o estándares que sirven de guía para proteger los recursos disponibles. Entre ellas se incluyen normas como ISO/IEC-27000 [fStIEC05], NIST-SP800 [oST07], CVSS [oIRT16] o MAGERIT [dHyAP12]; y plataformas como ITIL o COBIT [POK13]. Sin embargo sus bases han demostrado deficiencias al ser implementadas sobre escenarios dinámicos, donde el contexto juega un papel relevante a la hora de tomar decisiones [WAMS14]. Este es el caso de los entornos de monitorización de red, en especial aquellos que involucran

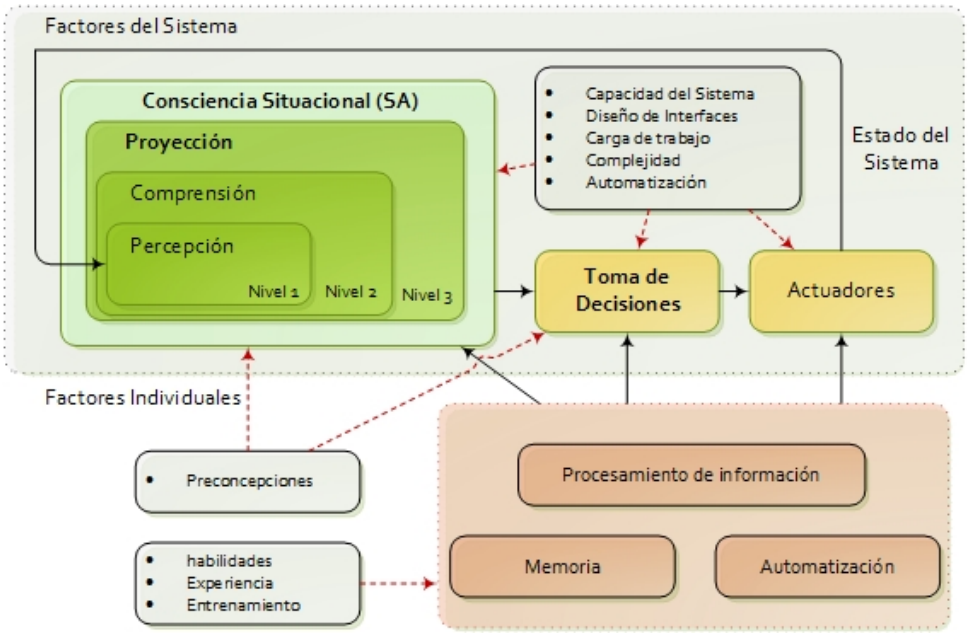

Figura 12.1: Consciencia situacional según el modelo de endsley

tecnologías 5G. Como solución a este problema algunos autores han adoptado metodologías de gestión de incidencias capaces de tratar la información de manera cognitiva, y que por lo tanto, facilitan su comprensión por medio del análisis contextual. De entre ellas destacan las que se basan en construir la Consciencia Situacional del entorno protegido mediante la aplicación del modelo de Ensdley, donde se estudia la percepción, comprensión y proyección del estado del sistema [End88]. La adaptación de este paradigma a la gestión de la seguridad en redes ha llevado a acuñar el término Seguridad en Redes basada en Consciencia Situacional (Del inglés NSSA) [LM15]. Sin embargo, a pesar de que se ha demostrado su eficacia en redes actuales, aún no ha sido adaptado a las dificultades que plantean los sistemas 5G. Esta sección describe dos conceptos: la gestión de incidencias y conciencia situacional.

12.1.1 Gestión de Riesgos

El problema de la gestión de riesgos es un tema de interés en la comunidad investigadora desde hace más de cuatro décadas. En consecuencia han sido publicados diversos trabajos que tratan de recopilar las contribuciones más relevantes, siendo [Ave16, SSABC16] algunas de las más actuales. En términos generales la bibliografía abarca una colección muy grande de tópicos que estudian desde la propia definición de riesgo y su planteamiento científico [HA14, Hol14], hasta el cómo son tratados a nivel gubernamental [Dot15]. La necesidad de gestionar la defensa de las tecnologías de la información ha dado pie a diferentes herramientas para guiar a las organizaciones a su implementación, incluyendo estándares [fStIEC05, oST07, dHyAP12] y plataformas [oIRT16, POK13]. La mayor parte de estas aproximaciones coinciden en que el proceso de gestión de incidencias debe recorrer las siguientes etapas: definición de riesgos, evaluación, monitorización y respuesta [SSABC16]. En la primera de ellas se lleva a cabo la delimitación de las situaciones

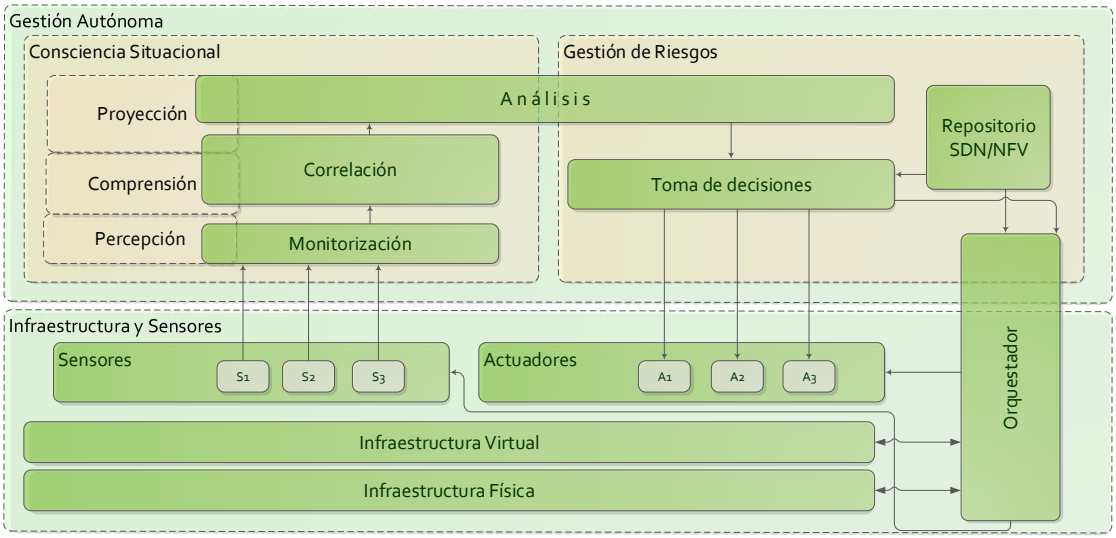

Figura 12.2: Arquitectura de gestión de riesgos para redes 5G

observables en el sistema con características potencialmente dañinas, teniendo en cuenta los objetivos de las organizaciones a proteger, sus políticas, riesgos tolerables y principios de actuación [fStIEC05]. A continuación se procede a la identificación de los posibles riesgos, su valoración y el planteamiento de métricas que permitan medir su impacto en el sistema [oIRT16, dHyAP12]. Por su complejidad esta es la fase con mayor presencia en la bibliografía [YKLA15], habiendo motivado el desarrollo de sistemas específicos para la Evaluación de Riesgos en la Seguridad de la Información (Del inglés *Information Security Risk Assessment* (ISRA)). En la etapa de monitorización se examina el entorno protegido en busca de indicios de riesgos. En el caso de detectarse alguno de ellos, tiene lugar la etapa de respuesta, en la que se lleva a cabo el despliegue de contramedidas. Debido a las grandes diferencias que existen entre los diferentes escenarios de monitorización, el éxito de la toma de decisiones depende directamente de los procesos anteriores y de su capacidad de adaptación a cada caso de uso. Los siguientes son ejemplos de metodologías para facilitar sus integración en ambientes más específicos: [KPH+15] para el control industrial, [NZGH16] para sistemas embebidos y [CBB+16] en SCADA.

12.1.2 Consciencia Situacional

Según Endsley, Consciencia Situacional (SA) significa «tener conocimiento del estado actual de un sistema, entender sus dinámicas, y ser capaces de predecir cambios» [End88]. Su modelo se divide en tres etapas (ver Figura. 12.1): percepción, comprensión y proyección; en la primera se llevan a cabo las labores de monitorización e identificación de incidencias, en la segunda su asociación y en la tercera se predice la evolución del estado del sistema. A partir de dichas etapas se deciden las acciones a realizar y su modus operandi. Nótese que en este modelo existe realimentación entre los niveles de actuación/decisión con la SA, de tal manera que los resultados obtenidos influyen en las decisiones a tomar, facilitándose el uso de técnicas de diagnóstico avanzadas [LM15]. Este modelo ha sido implementado en diversas áreas, como gestión de incidencias en redes

-eléctricas inteligentes [DAKM15], generación de energía [NNL15] o sistemas para evitar colisiones de vehículos [MPTSF16]. En propuestas como [SV15] se plantea su adaptación a situaciones críticas por medio de distribución y priorización del tratamiento de datos. Tal y como se discute en [CSD15], el modelo de Endsley ha demostrado ser efectivo en escenarios complejos y dinámicos, donde el diagnóstico tiene una alta dependencia del contexto en el que se han reportado las incidencias [WAMS14]. En la seguridad de la información también ha jugado un papel esencial [FB14], con un claro predominio de las implementaciones para la gestión de riesgos en situaciones de emergencia, sistemas industriales y redes. En ellas son mejoradas tres de las deficiencias de los sistemas para la Gestión de Riesgos en la Seguridad de la Información (Del inglés ISRM) más repetidas en la bibliografía: no aprovechar todas las posibles fuentes de información, estimación de riesgos sin tener suficientemente en cuenta el contexto en que son registrados o su proyección, y la dificultad de llevar a cabo procesos de auditoría continuos [SM10]. Para tratar de dar solución a estos problemas sin perder la perspectiva aportada por los ISRM/ISRA, algunos trabajos han combinado ambos paradigmas, tal y como se observa en [WAMS14, NLZ14], donde la Consciencia Situacional se construye teniendo en cuenta la definición de riesgos y su evaluación al aplicar directivas.

12.2 Gestión de Incidencias en 5G

Los principios de diseño de las nuevas plataformas móviles 5G tienen como objetivo el soporte de un incremento exponencial de dispositivos conectados y del tráfico que circula por la red. El soporte en tiempo real de estos servicios requiere, a diferencia de arquitecturas rígidas tradicionales, eliminar la estrecha unión hardware/software propietario y permitir una visión global junto con configuración y actualización dinámica de las diferentes operaciones de la red. Con este objetivo, modelos de diseño basados en tecnologías SDN/NFV, utilizadas inicialmente en redes de datos cableadas, han sido extendidos a plataformas inalámbricas y móviles. De esta manera, los operadores pueden gestionar la infraestructura móvil evitando la configuración manual, individual y remota de los diferentes equipos (generalmente utilizando línea de comandos CLI). Sin embargo, la gestión automática de riesgos en plataformas móviles que aprovechen este nuevo paradigma es escasa o prácticamente nula. El reto primordial es la coordinación entre los dispositivos repartidos en diferentes puntos de la infraestructura y las funciones virtuales que pueden ser instanciadas dinámicamente. De igual manera, dichos avances están limitados por la falta de esquemas que faciliten el procesamiento de altas cantidades de información que sirvan para detectar problemas en la red, y en la forma de analizar las causas de dichos problemas. Es claro que los diferentes eventos o incidencias tienen que ser organizados y priorizados adecuadamente de tal manera que no comprometan la seguridad de la información y la calidad de servicio que circula por la red.

12.3 Arquitectura para la Seguridad de la Información en Redes 5G

En la arquitectura propuesta, el análisis situacional tiene como objetivo la lectura del estado actual de los elementos monitorizados (infraestructura de red) y en caso de ser necesario, la respuesta automática a problemas de red identificados de manera reactiva o preventiva. Con este fin, se han identificado los siguientes requerimientos y supuestos:

- Los elementos de infraestructura monitorizados son compatibles con la tecnología SDN/NFV. En el caso de existir elementos de hardware/software no compatibles, se considera una capa de compatibilidad, la cual emula el funcionamiento de un elemento virtualizado compatible.
- La comunicación entre los diferentes elementos de la arquitectura se desarrollan por medio de canales seguros.
- La información proveniente de los elementos de monitorización (métricas, alertas) se consideran confiables.
- Los procesos de Análisis Situacional se ejecutan de manera independiente y transparente. Es decir, estas operaciones no afectan el rendimiento de los elementos de la infraestructura 5G.
- Los módulos funcionales representan tareas que en la práctica pueden ser implementados en arquitecturas distribuidas según las necesidades de velocidad de conexión y capacidad de cómputo.

En la Figura. 12.2 se ilustra la distribución de los diferentes módulos y la sinergia entre la arquitectura propuesta y el modelo de Endsley. Se han definido cuatro niveles lógicos principales: Infraestructura Virtual y Sensores, Monitorización/Correlación, Análisis y Decisión/Actuadores, donde el primero abarca las tareas relacionadas con la percepción, el segundo las de comprensión y el tercero las de decisión y ejecución de contramedidas. Por lo tanto Infraestructura/Sensores y Monitorización/Análisis construyen el SA, debiendo mantener una realimentación con Decisión/Actuadores que permita hacer el seguimiento de diagnóstico. A continuación se describe el rol de cada componente en la arquitectura.

12.3.1 Infraestructura Virtual y Sensores

El principal objetivo de esta capa de procesamiento de datos es dar soporte al despliegue de los elementos necesarios para la captura de la información requerida para inferir riesgos del sistema. Su despliegue aprovecha uno de los principios de diseño de la nueva arquitectura móvil 5G: capacidad de integración con ambientes virtuales y la nube. De este modo se facilita el despliegue dinámico de elementos de red, promoviéndose el desarrollo de una capa de infraestructura completamente virtualizada: todos los elementos físicos o hardware tales como estaciones base, enlaces, encaminadores o servidores son gestionados por una capa de virtualización. A este nivel la tecnología SDN desacopla los planos de datos y de control

de cada uno de los dispositivos de red. El usuario puede crear aplicaciones de software para modificar dinámicamente el comportamiento del plano de datos. Por su parte, NFV permite el tratamiento de las diferentes funciones de red (cortafuegos, DPI, balanceador) como funciones de software independientes del equipo. De esta manera, se consideran las aplicaciones tipo SDN (SDN-Apps) o NFV (NFV-Apps). En la arquitectura propuesta, los sensores son un tipo de NFV-Apps encargados de la monitorización de diferentes métricas del sistema. Ejemplos de sensores pueden ser: analizadores de tráfico, detectores de anomalías, monitores de la calidad de servicio (QoS/QoE), etc. Al ejecutarse sobre un entorno virtualizado, los sensores (NFV-Apps) pueden cambiar dinámicamente su posición y características de monitorización. Esto permite aumentar la vigilancia en las regiones que están siendo atacadas, y delimitar zonas de cuarentena. Además facilita la realización de cambios en los parámetros a observar, dando pie a la posibilidad de considerar elementos de monitorización de propósito general adaptables a las circunstancias de la red.

12.3.2 Monitorización y Correlación

La capa de monitorización recoge la información proveniente de los niveles inferiores (infraestructura virtual y sensores) y aplica técnicas de correlación para simplificar su análisis. Por lo tanto cuenta con dos componentes: extracción de datos y correlación. A continuación se describe cada uno de ellos:

- *Monitorización.* Los principales objetivos de la monitorización son recopilar y gestionar información proveniente de todas las fuentes de información, y facilitar su acceso a capas superiores. Este módulo también gestiona el registro y acceso de nuevos sensores. La información recabada es organizada en estructuras de datos eficientes tomando en cuenta la alta cantidad de información a procesar. En este sentido se han considerado dos escenarios: en el primero de ellos el sensor envía un reporte al monitor cuando encuentra información considerada importante (alertas, caída de un enlace, sobrecarga de memoria o CPU); en el otro escenario, cuando el monitor considere oportuno puede solicitar al sensor información necesaria para las tareas de agregación o análisis (topología virtual, enlaces libres, entre otros).

- *Correlación.* Se encarga del primer nivel de abstracción del procesamiento de información, en el cual, con el objetivo de tener una visión global del estado de la red, se ejecutan procesos de correlación y agregación. La información considerada redundante o no sensitiva es descartada. Es decir, por ejemplo, en el caso de recibir múltiples alertas provenientes de cada dispositivo de una misma zona afectada, se indica una única alerta junto con la topología afectada. Debido al dinamismo que ofrecen los ambientes virtuales en contraste con la rigidez de los elementos físicos, la topología se encuentra expresada como un grafo extendido o aumentado ($G_a(V_a, E_a)$), el cual modela los nodos (V_a) y enlaces (E_a) virtuales localizados en la infraestructura física [SKW+15, CRB09]. Asimismo, el resultado de operaciones de correlación y agregación permiten que las métricas recibidas a bajo nivel puedan ser expresadas o traducidas en métricas de alto nivel, también conocidas como estado

de red (Del inglés HoN). Por ejemplo, la tasa de transmisión (Mbps), retardo (ms) y jitter (ms) de un flujo de datos de video *streaming*, recibido por los sensores en diferentes puntos de la red, puede ser expresado como una percepción global de la calidad de servicio QoS/QoE, cuantificada mediante la medición del MOS.

12.3.3 Análisis

En el componente de análisis se lleva a cabo la identificación de situaciones de red a partir de las métricas recibidas desde el módulo de correlación y se construyen diagnósticos que facilitarán la decisión de las contramedidas a aplicar. En ella se distinguen dos tipos de situaciones: *eventos* y *riesgos*. Los *eventos* son incidencias que a priori no presentan naturaleza dañina, pero que sin embargo pueden mejorar el diagnóstico y la valoración de riesgos. Ejemplos de eventos son la detección de nuevos dispositivos de red, identificación de dispositivos inactivos o el despliegue de nuevas capas de virtualización. Por otro lado, los *riesgos* son situaciones que directamente conllevan vulneraciones en la seguridad del entorno protegido, como la explotación de vulnerabilidades, ataques de denegación de servicio o accesos no autorizados. Pueden ser inferidos a partir de eventos u otros riesgos. En Figura. 12.3 se ilustran las principales etapas del proceso de análisis: detección, identificación y evaluación de riesgos, gestión del inventario de activos, construcción de mapa de riesgos, predicción, diagnóstico y seguimiento de contramedidas. A continuación se describe brevemente cada una de ellas.

- *Detección.* Enlace entre los módulos de monitorización con las funciones de comprensión de la información. Tiene como datos de entrada las métricas de alto nivel construidas a partir de datos agregados y detecta las posibles situaciones inferibles a partir de ellos.

- *Identificación y evaluación de riesgos.* Implementación de normativas y/o plataformas para la identificación y evaluación de riesgos. Este componente puede formar parte de las funciones del módulo de detección o actuar de manera independiente. Dada la taxonomía propuesta en [SSABC16] y las características de 5G, es recomendable que se consideren criterios de evaluación cualitativos desde una perspectiva basada en servicios, teniendo en cuenta la propagación de las incidencias a lo largo de la red.

- *Inventario de activos.* Las redes 5G tienen la capacidad de automatizar el despliegue de nuevos servicios y dispositivos de red en función de su estado, situación que implica gran dificultad a la hora de medir el impacto de los riesgos. Con el fin de contribuir a su desarrollo, este módulo se encarga de la construcción y mantenimiento del inventario de activos del sistema.

- *Mapa de riesgos.* Con el fin de facilitar las tareas de diagnóstico y toma de decisiones, en este componente se genera y gestiona un mapa de riesgos de la red. En su construcción participan las métricas correlacionadas ofrecidas por las capas de percepción y las situaciones detectadas a nivel de análisis.

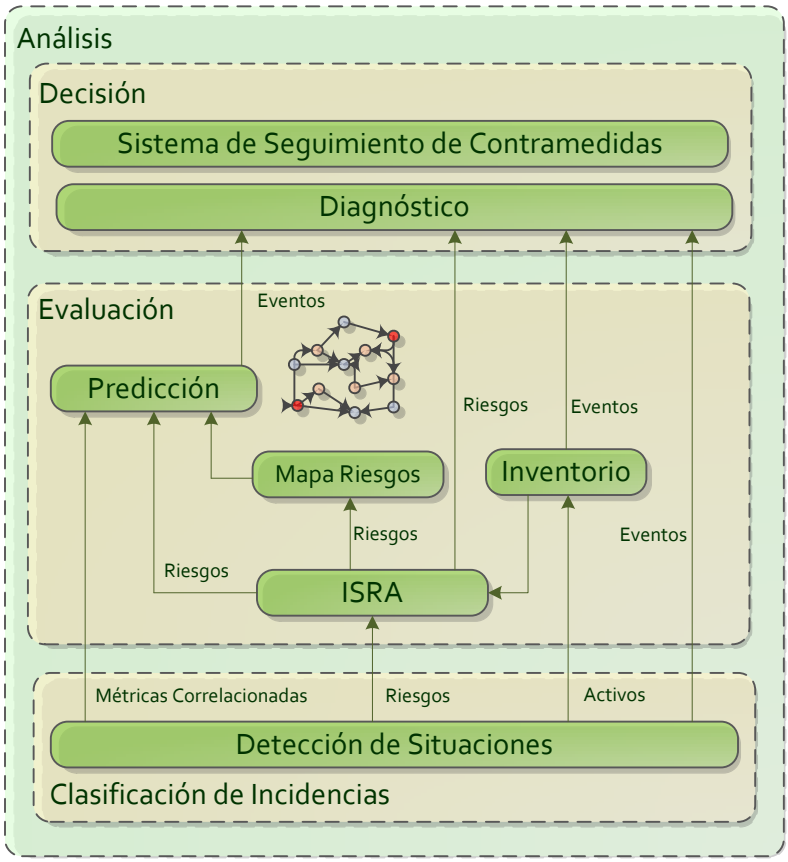

Figura 12.3: Análisis de situaciones en 5G

- *Predicción.* Proyección de la Consciencia Situacional en el modelo de Endsley. A partir del mapa de riesgos y las incidencias identificadas, este componente aplica algoritmos para anticipar cambios en la red.

- *Diagnóstico.* Análisis de alto nivel de riesgos, su valoración, impacto, proyección y estado de la red con el fin de identificar situaciones de mayor complejidad poco visibles en los niveles inferiores. Por ejemplo, el componente de diagnóstico puede reconocer botnets por medio de la relación de riesgos que determinan la presencia de dispositivos de red infectados y tráfico anómalo en sus proximidades.

- *Seguimiento de Contramedidas.* Principal enlace entre las tareas de análisis con las de toma de decisiones. El sistema de seguimiento de contramedidas comunica los problemas diagnosticados a las siguientes capas de procesamiento de información y construye un historial con las acciones ejecutadas para su mitigación. Según el modelo de Endsley, forma parte de la realimentación entre actuadores y comprensión. Además mejora las tareas de diagnóstico, facilita la realización de cambios en las decisiones a tomar en función de los resultados de situaciones previas similares.

12.3.4 Toma de Decisiones y Actuadores

La toma de decisiones busca mitigar los problemas que afectan la normal operación de los elementos de la red y, de ser el caso, optimizar el rendimiento de los diferentes servicios que se brindan. Con este objetivo, el sistema recibe la información proveniente de la etapa de análisis y selecciona un conjunto de acciones o respuestas a ejecutarse. Las acciones disponibles se encuentran repartidas entre las diferentes funciones NFV-Apps. Por ejemplo, en respuesta a un ataque de denegación de servicio, el sistema utiliza el informe recibido por análisis y toma la decisión de instalar funciones de firewall en puntos estratégicos. De igual manera, la información proveniente de análisis puede incluir resultados de algoritmos de predicción. Estos datos sirven para ejecutar acciones de manera proactiva, es decir, evitar o disminuir la probabilidad de que los servicios se vean afectados negativamente. Cuando el número de usuarios conectados a un servicio se incrementa paulatinamente y el análisis determina que el tráfico es legítimo y pronostica un incremento de tráfico, el sistema puede automáticamente instanciar balanceadores de carga para evitar un futuro colapso del servicio. La ejecución de las diferentes acciones es coordinada por un agente orquestador, el cual se asegura que los recursos virtuales para aplicar dichas acciones se encuentran disponibles y no afectarán el rendimiento del sistema.

12.4 Resumen del Capítulo

Este capítulo introduce una arquitectura para la gestión de incidencias en redes 5G. La propuesta combina las bases de los esquemas de gestión de riesgos tradicionales con el modelo de Consciencia Situacional publicado por Endsley. En ella se cubren todos los niveles de procesamiento de información de las redes 5G, desde su infraestructura hasta los actuadores encargados de aplicar las acciones de mitigación. Además, este capítulo proporciona una idea general de las tareas de identificación, monitorización, análisis, toma de decisiones y predicción.

Capítulo 13

Marco de Análisis de SELFNET

Este capítulo presenta el diseño del Módulo de Análisis de SELFNET, el cual tiene como objetivo principal identificar situaciones inesperadas o sospechosas basadas en métricas proporcionadas por sensores y diferentes componentes de red. El módulo de Análisis de SELFNET provee una arquitectura basada en casos de usos donde las funciones de análisis pueden ser extendidas fácilmente. El resto de este capítulo se encuentra estructurado de la siguiente forma. La sección Section 13.1 describe las características principales y las capacidades de análisis del proyecto SELFNET y su relación con el modelo de conciencia situacional. La sección 13.2 detalla los principios de diseño, los requerimientos y la arquitectura del Marco de Análisis en su totalidad. La sección 13.3 muestra este módulo como una caja negra, enfatizando sus entradas y sus salidas. La sección 13.4 define la especificación de datos de los descriptores para los casos de uso. La sección 13.5 ilustra algunos ejemplos de la especificación de datos y sus flujos de trabajo. Finalmente, la sección 13.6 resume este capítulo.

13.1 Módulo de Análisis vs Modelo de Conciencia Situacional

El Proyecto H2020 SELFNET [sel17] tiene como objetivo proveer un marco de gestión autónomo de red para infraestructuras 5G a través de la integración de tecnologías novedosas tales como SDN, NFV, SON, computación en la nube e inteligencia artificial. SELFNET permite la mitigación autónoma de problemas existentes o potenciales a través de acciones correctivas y preventivas, mientras provee escalabilidad, extensibilidad y la reducción de los costos de capital (capex) y de operación (opex). Estas capacidades son provistas por medio de una arquitectura en capas y un enfoque basado en casos de uso, como es detallado en [NCC+16]. La arquitectura de SELFNET cubre los problemas principales de la gestión de red, tal es el caso de capacidades de auto protección contra ataques cibernéticos distribuidos, capacidades de auto reparación contra fallos de red y capacidades de auto optimización para mejorar dinámicamente el rendimiento de la red y la calidad de experiencia de los usuarios.

Para ello, SELFNET define dos tipos de funciones avanzadas de red: i) sensores para la monitorización de información específica y ii) actuadores para mitigar problemas

potenciales. En particular la inteligencia de red es provista por la capa Autónoma de Red. Esta capa recopila métricas relacionadas al comportamiento de la red y utiliza ésta información para inferir el estado de la misma. Luego, decide las acciones a ser ejecutadas para cumplir los objetivos del sistema. La capa autónoma SON está compuesta por dos subcapas: i) Subcapa de Monitorización y Análisis y ii) Subcapa de Gestión Autónoma. La subcapa de Monitorización y Análisis se basa en los principios del modelo de conciencia situacional de Endsley y está conformada por tres módulos: monitorización y descubrimiento, agregación y correlación y el módulo de Análisis. Estos 3 módulos se relacionan con las funciones de percepción, comprensión y proyección del modelo de Endsley, como se muestra en la figura. 13.1.

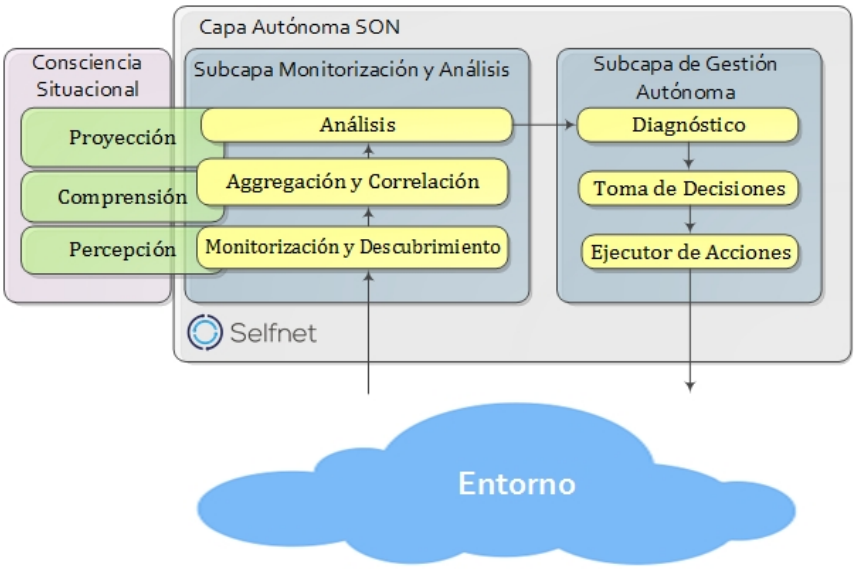

Figura 13.1: Endsley vs. capa autónoma de SELFNET

El principal objetivo del módulo de Análisis es inferir información de las métricas monitorizadas para facilitar respuestas reactivas y proactivas sobre la infraestructura de red (ej. Mejorar las tareas de diagnóstico y la toma de decisiones). Por la tanto el Módulo de Análisis es el primer paso para proveer inteligencia al sistema, donde complejas conclusiones son inferidas a partir del razonamiento de la información provista por las fases de monitorización y agregación y la definición de cada caso de uso. Por tal motivo, el módulo de análisis distingue tres grandes tareas para el procesamiento de la información: reconocimiento de patrones, razonamiento y predicción. Las conclusiones alcanzadas son representadas en forma de síntomas relacionados con cada caso de uso. Teniendo esto en mente, es posible afirmar que el módulo de Análisis de SELFNET provee un modelo de conciencia situacional orientado a síntomas en concordancia con las situaciones definidas por cada caso de uso.

13.2 Módulo de Análisis de SELFNET

En esta sección el diseño del módulo de Análisis de SELFNET es detallado. Se describe también las consideraciones iniciales, los requerimientos, los principios de diseño y la arquitectura de dicho módulo.

13.2.1 Consideraciones y Requerimientos Iniciales

Esta subsección describe los requerimientos más relevantes y las consideraciones iniciales tomadas en cuenta en el diseño del Módulo de Análisis:

- *Escalabilidad.* La propuesta debe permitir añadir capacidades nuevas (extensibilidad) acorde a los principios de diseño de SELFNET. Por tal razón, la integración de funciones de análisis adicionales será realizada mediante plugins.

- *Basado en casos de uso.* Dada la gran dependencia de las tareas de análisis con respecto a los casos de uso, la definición básica de las observaciones a ser estudiadas (objetos de la base de conocimiento, reglas, métricas de predicción, etc.) son provistas por el operador del caso de uso. En consecuencia, el módulo de análisis es escalable a diferentes contextos.

- *Adquisición de Conocimiento.* Es ampliamente conocido que la desventaja más común de un sistema experto es el problema de adquisición de conocimiento. Por consiguiente, tener operadores cualificados que puedan especificar adecuadamente las reglas de los casos de uso, no es siempre sencillo. El módulo de Análisis de SELFNET no aborda los problemas relacionados con la adquisición de conocimiento. Se asume que dicho conocimiento es definido por operadores calificados en cada caso de uso o por algoritmos de aprendizaje automático.

- *Definición de reglas amigable.* La definición de reglas es un asunto complicado, en el cual inclusive un operador cualificado puede cometer errores de ambigüedad o coherencia. Para mitigar dichos problemas, la configuración y definición de nuevos casos de uso y sus reglas, deberá realizarse de forma amigable.

- *Incertidumbre.* La lógica clásica permite solamente un razonamiento exacto. Se asume que el conocimiento perfecto siempre existe, sin embargo esto se encuentra lejos de la realidad de SELFNET. Para mejorar la calidad de las conclusiones, el módulo de Análisis gestiona el conocimiento teniendo en mente el concepto de incertidumbre. Este es particularmente apropiado para determinadas características de análisis, tal como el estudio de observaciones basado en umbrales de decisión o intervalos de confianza.

- *Filtrado.* Inicialmente, el filtrado de los síntomas no es considerado. Por tanto cada síntoma inferido, sin importar su naturaleza o incertidumbre, es transmitido a la etapa de diagnóstico/toma de decisiones, donde su impacto y relevancia son evaluados apropiadamente.

13.2.2 Principios de Diseño

Los siguientes principios de diseño y limitaciones sientas las bases del Marco de Análisis y sus componentes.

- *Big Data.* Para tratar con conjuntos de datos grandes y homogéneos, big data provee algoritmos de predicción, análisis de la conducta de los usuarios y funcionalidades para agregación/correlación [SKIW17]. Estas capacidades son principalmente tomadas en cuenta en las tareas de monitorización y agregación. El módulo de Análisis utiliza métricas agregadas y correlacionadas, por tanto se reduce la cantidad de información a ser analizada. En el enfoque presentado, la implementación de tecnologías big data para gestionar toda esta información es opcional.

- *Entorno de monitorización estacionario.* De acuerdo a Holte et al. [Hol93], en un entorno de monitorización estacionario, las características y la distribución de las observaciones normales a ser analizadas se relacionan con las muestras de referencia del proceso de aprendizaje. Si la distribución del entorno de monitorización cambia representativamente, es considerado no estacionario. Otro problema que reduce la calidad del proceso de análisis es la presencia de cambios graduales, a lo largo del tiempo, en las estadísticas de la clase a la cual pertenece la observación. En la literatura esta fluctuación es conocida como "concept-drift" y es ampliamente discutida en [DRAP15]. El módulo de Análisis al operar en un entorno estacionario brinda una solución simple y eficiente, sin embargo es propensa a pequeños fallos cuando ocurren cambios. Por otra parte, si se considera un entorno de monitorización no estacionario se mejora la precisión, pero implica nuevos retos tales como la detección de cambios, técnicas de regresión, la identificación de cuando la calibración debe ser completada, o la selección de las muestras a ser tomadas en cuenta en los nuevos entrenamientos. Dada la complejidad que implica, el módulo de Análisis asume un entorno de monitorización estacionario.

- *Datos de Alta Dimensión.* El análisis de "datos de alta dimensión" implica considerar datos cuya dimensión es mayor a las dimensiones del análisis multivariante clásico. Como se indica en Bouveyron et al. [BBS14], cuando los métodos tradicionales tratan con datos de alta dimensión son susceptibles a sufrir la bien conocida "maldición de la dimensión", donde el gran número de atributos irrelevantes o redundantes a ser considerados conducen a errores importantes de predicción. Por tanto, trabajar con estos datos implica la necesidad de algoritmos más complejos y específicos. En términos de SELFNET significa que el vector de las métricas de salud de la red (del inglés HoN) es lo suficientemente grande para que la implementación de métodos específicos sean considerados en la optimización de las tareas de procesamiento. A priori no existen indicios de que el procesamiento requerido por SELFNET sea de este tipo. Por tanto, este trabajo no diferencia datos convencionales de los datos de alta dimensión, asumiendo que la tarea de agregación es capaz de optimizar la cantidad de atributos a ser analizados.

- *Supervisión.* Modo de entrenamiento de SELFNET. Los métodos de análisis basados en modelado/regresión asumen que el conocimiento puede ser inferido de las observaciones de una fase de aprendizaje previa. El proceso de aprendizaje con frecuencia requiere datos de referencia los cuales permiten identificar los rasgos característicos del entorno de monitorización tal como reglas, límites, matrices de incidencia, vectores de dirección o estadística básica. Debido a la complejidad de diseñar un modo de entrenamiento, este trabajo describe como se obtiene la información necesaria para la construcción de nuevos modelos.

- *Diseño centralizado.* Para asumir un enfoque centralizado es necesario proveer un esquema de propósito general donde la integración de nuevos casos de uso sea completamente configurable por especificación y no requiera la actualización en la implementación (Ver figura 13.2). Por tanto, un enfoque centralizado no depende de las características del caso de uso y en consecuencia es altamente escalable y eficiente (Evita redundancia). Sin embargo, su diseño y la descripción de los casos de uso es una tarea compleja. Por otra parte, un enfoque distribuido incluye un componente adicional por cada caso de uso en el cual métodos específicos de reconocimiento de patrones y predicción son implementados a través de plugins. Los mecanismos de pre procesamiento, selección y descubrimiento de síntomas son de propósito general. En esencia, un enfoque distribuido es fácil de diseñar pero es dependiente del caso de uso; cada vez que un nuevo caso de uso es incorporado, la implementación deberá ser actualizada. Debido al alto impacto en la escalabilidad de un diseño distribuido, el módulo de Análisis de SELFNET considera un enfoque centralizado.

- *Encapsulación de Datos.* Uno de los grandes retos al diseñar el módulo de Análisis de SELFNET es el tratamiento de datos desconocidos. Es posible asumir que los casos de uso no proveen información lo suficientemente clara para ser analizada (de hecho, los casos de uso futuros son completamente desconocidos). En la fase de especificación, el operador del caso de uso tiende a brindar información cualitativa (de buena calidad) de las métricas a considerar, sin embargo puede pasar por alto detalles acerca de su naturaleza cuantitativa (ej. tipo de dato, dominio, rango, restricciones, etc.), la cual debería ser considerada en las tareas de análisis. Además que la información cuantitativa es mucho más dependiente del caso de uso. Con el objetivo de restar relevancia a los detalles cuantitativos (lo cual es el núcleo de las tareas de agregación y correlación), y para facilitar la incorporación de nuevos casos de uso a través de descriptores de propósito general, el módulo de Análisis de SELFNET está basado en la encapsulación de datos en dos niveles de abstracción: parámetros cuantitativos y cualitativos (Figura 13.3). La primera es independiente de los casos de uso y permite el diseño de un marco de análisis centralizado válido para cualquier tipo de especificación de datos. Por su parte, los parámetros cualitativos recolectan información estrechamente relacionada con el caso de uso al que pertenecen (nombre de las métricas, fuente, ubicación, tenant, etc.). Estos datos son requeridos principalmente por las tareas de agregación/correlación, diagnóstico y toma de decisiones.

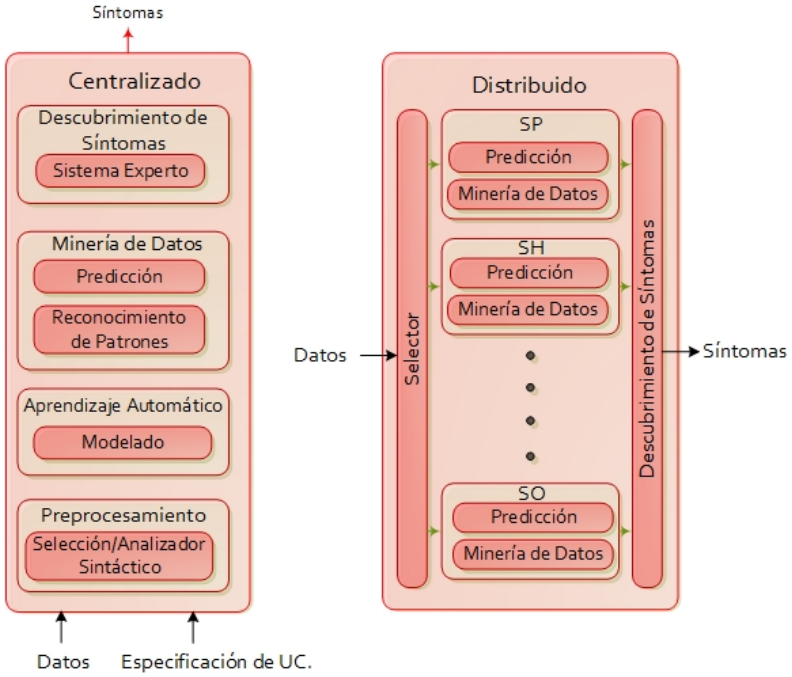

Figura 13.2: Arquitectura centralizada vs. distribuida

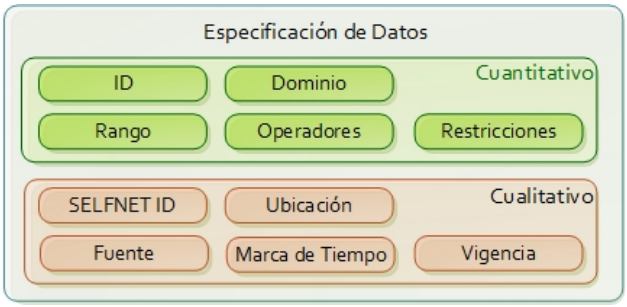

Figura 13.3: Ejemplo de la encapsulación de datos

13.2.3 Arquitectura del Módulo de Análisis

En la figura. 13.2 la arquitectura del módulo de Análisis es ilustrada. Se distingue los siguientes ocho componentes: (1) Reconocimiento de Patrones, (2) Predicción, (3) Umbrales Adaptativos, (4) Base de Conocimiento, (5) Motor de Inferencia, (6) Memoria, (7) Interfaz de Usuario y (8) Estimación de Incertidumbre. El conjunto (4),(5),(6),(8) está relacionado con el razonamiento, (1),(2),(3) con proyección y (7) con la administración de los casos de uso. Las tareas son resumidas de la siguiente forma:

- *Reconocimientos Patrones.* El reconocimiento de patrones relaciona los patrones adquiridos o conocidos con anterioridad con los datos agregados (i.e. $Fa(T_h), Fa(KPI), Fa(Ev)$), y retorna los Hechos Fa con los resultados de su estudio. Con este propósito, diferentes tareas internas son ejecutadas: estudio de los datos de entrada (tanto los datos de entrenamiento como las muestras a ser analizadas), decisión de la estrategia de datos más adecuada para cada contexto, características

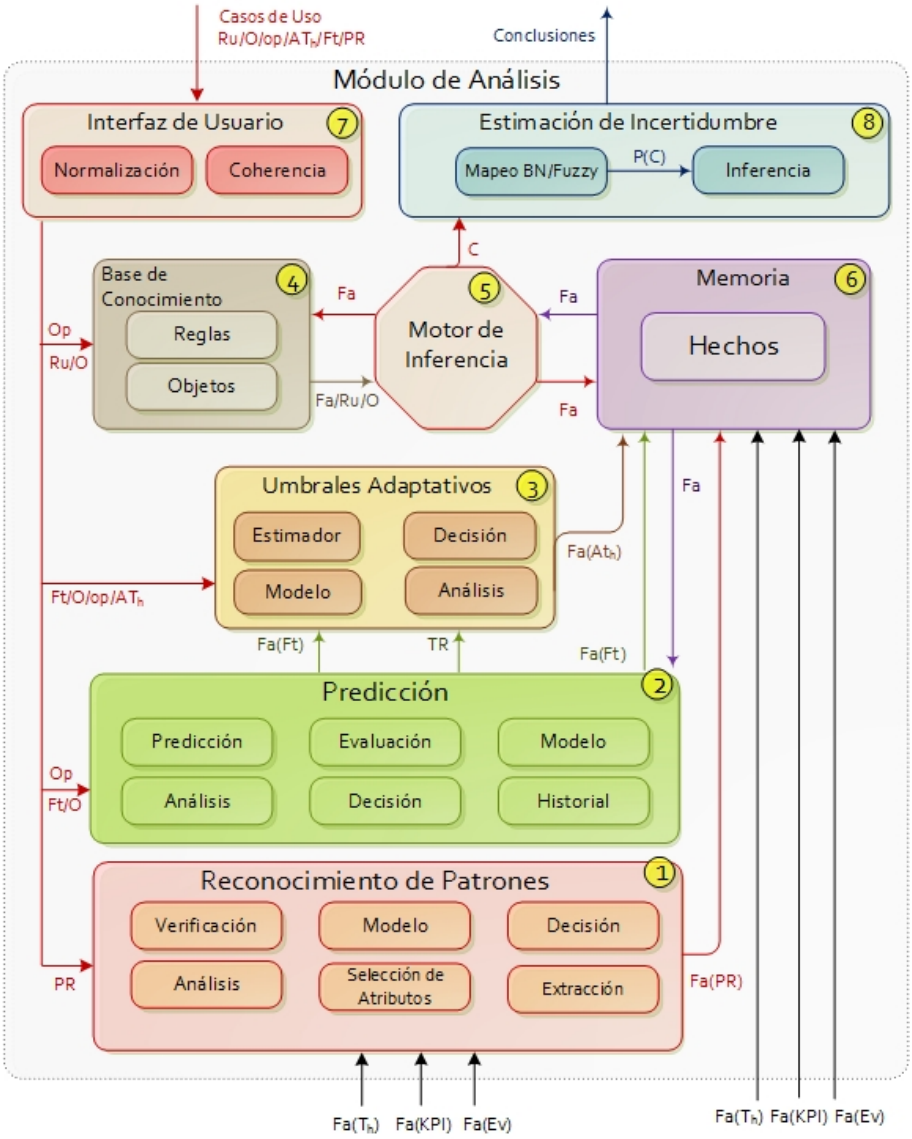

Figura 13.4: Arquitectura del módulo de análisis

de extracción, construcción de modelos/regresiones, análisis de hechos relacionados con los datos agregados. La bibliografía recolecta una gran cantidad de métodos de reconocimiento de patrones, los cuales son adaptados a las necesidades de los casos de uso y a las características de los diferentes ambientes de monitorización [DSBA16]. El módulo de Análisis de SELFNET se enfoca en dos acciones fundamentales: la identificación de firmas de eventos conocidos previamente [MLK14] y la detección de anomalías [Agg15].

- *Componente de Predicción.* Este componente calcula las métricas de predicción (como Hechos Fa) asociadas a cada caso de uso a partir de las observaciones provistas por la fase de agregación (Umbrales T_H, Indicadores de Rendimiento KPI y Eventos Ev). Este proceso implica diferentes pasos de procesamiento: gestión del historial con los datos requeridos para construir el modelo de predicción, análisis de las características

relevantes para decidir el algoritmo de predicción más adecuado y la evaluación de los resultados que facilitan el aprendizaje basado en las decisiones previas. Como se indica en [KD15], la predicción de eventos de red mejoran la optimización de recursos, el despliegue de acciones proactivas y anticipa la identificación de riesgos. El módulo de Análisis de SELFNET se enfoca principalmente en inferir predicciones de dos estructuras de datos: series temporales y grafos. El primero ayuda a determinar la evolución de las métricas HoN, por consiguiente implementa principalmente algoritmos de suavizamiento exponencial (del inglés exponential smoothing) [GD80] y modelos auto-regresivos [KHC+16]. Por otra parte, la evolución de grafos es utilizada para anticipar el descubrimiento de nuevos elementos [BPC13] y facilitar la gestión de recursos [RG12].

- *Umbrales Adaptativos.* Establece medidas para aproximar cuando los errores de predicción deben ser tomados en cuenta en la identificación de síntomas. Por consiguiente recibe como parámetros de entrada los valores relacionados con las métricas de predicción (registro de seguimiento TR y Predicción Ft), y devuelve los umbrales adaptativos AT_h. Su construcción implica diferentes pasos, tales como el análisis y la extracción de las características principales de los datos de entrada, decisión de los algoritmos más adecuados, modelado y estimación de umbrales. El módulo de Análisis de SELFNET construye umbrales adaptativos a partir de los datos representados como series temporales o grafos, los cuales permiten inferir conclusiones más precisas a partir de cada estimación generada por el componente de predicción. La principal aplicabilidad de los umbrales adaptativos es que considera el contexto del entorno de monitorización en la inferencia de nuevos hechos relacionados con el filtrado [ZWH+16], y reduciendo la tasa de falsos positivos [BFK+17].

- *Base de conocimiento.* Guarda la información específica de cada caso de uso. Estos datos son representados por objetos y reglas. Los objetos O son la unidad básica de información (ej. Temperatura, congestión, latencia, etc). Las reglas Ru son las directrices que permiten la inferencia de hechos y conclusiones. Los hechos, objetos y sus valores son relacionados a través de las operaciones. A priori, en este enfoque el aprendizaje máquina en línea no es considerado para adquirir conocimiento de los casos de uso en tiempo real [VE16], tal como la definición de nuevas reglas, priorización de las métricas, etc. (Ej. la información a ser considerada parte del entrenamiento original).

- *Motor de inferencia.* Aplica las reglas Ru a la base de conocimiento con el objetivo de deducir nuevo conocimiento. Este proceso se repetirá a medida que cada nuevo hecho Fa de la base de conocimiento pueda desencadenar reglas adicionales. Tradicionalmente, el motor de inferencia opera en uno de los siguientes dos modos: forward chaining and backward chaining [HRWL84]. Inicialmente, el primero considera los hechos conocidos con anterioridad e infiere nuevos hechos. Por otro lado, backward chaining considera los hechos e infiere cuáles fueron sus causas. Puesto que el módulo de Análisis de SELFNET infiere conclusiones a partir de hechos descubiertos, el primer enfoque es implementado. Además, es importante tener en cuenta que la implementación más sencilla del motor de inferencia considera reglas de separación

(reglas modus ponens) basadas en lógica proposicional [MMRAT16]. Las reglas pueden ser adaptadas a una representación diferente de incertidumbre, tal como la lógica difusa [MF02], conjuntos aproximados (rough sets) [CLL+15] o redes Bayesianas [FN01]. Para facilitar el entendimiento de esta propuesta, la especificación de reglas en el módulo de Análisis de SELFNET aplica solamente reglas básicas de lógica proposicional (como se describe en la sección 13.4).

- *Memoria.* Almacena todos los hechos conocidos (Fa) de los casos de uso (ex. $Temperature = 3°$, $Latency > 200ms$, etc.), considerando los hechos inferidos ($Fa(PR)$, $Fa(AT_h)$, $Fa(Ft)$) y aquellos provistos por las fases de monitorización y agregación de SELFNET ($Fa(T_h)$, $Fa(KPI)$, $Fa(Ev)$). Los metadatos relacionados con la información cuantitativa de la naturaleza de los hechos descubiertos también son almacenados.

- *Interfaz de Usuario.* Configura el reconocimiento de patrones PR por cada caso de uso y permite actualizar la base de conocimiento insertando, modificando y eliminando datos asociados con cada caso de uso, tal como los objetos O, reglas Ru, operaciones Op o las métricas de predicción Ft. La información es pre procesada para asegurar compatibilidad y coherencia [Gil12], evitando de ésta manera contradicciones y ambigüedad entre reglas.

- *Estimación de incertidumbre.* Complementa el motor de inferencia y facilita el estudio de las conclusiones teniendo en cuenta su incertidumbre. Sus entradas son las conclusiones obtenidas como síntomas potenciales de incidencias relevantes y la información asociada con su inferencia (hechos, desencadenadores, reglas, etc.). Este es el único elemento opcional de la arquitectura, puesto que su uso será solamente requerido cuando la tarea de diagnóstico de SELFNET [EN517] necesite evitar ambigüedad en las conclusiones o convertir la lógica del módulo de Análisis a un formato específico. Por ejemplo, cuando el motor de inferencia opera con reglas de lógica difusa, el elemento de estimación de incertidumbre genera un salida amigable para el módulo de diagnóstico basado en lógica clásica (del inglés crisp logic) [TC17].

13.3 Entradas/Salidas de Análisis

Si se estudia el módulo de Análisis como una caja negra es posible enfocarse en las entradas/salidas y su relación con el resto de los componentes de SELFNET [NCC+16]. Bajo esta perspectiva, se describe las fuentes de información, la naturaleza de los datos y su comportamiento en diferentes circunstancias. Como se muestra en la figura 13.5, el módulo de Análisis depende de tres fuentes de información, dos externas: el componente de agregación de SELFNET y el operador del caso de uso; y una fuente interna que corresponde a los datos generados por el propio módulo de Análisis. Las conclusiones son reportadas al módulo de Diagnóstico de SELFNET [EN517] como síntomas. El rol que juega cada uno de estos elementos es detalla a continuación:

- *Agregación.* Las observaciones llegan al módulo de Análisis a través de la capa de

agregación (Capacidades de percepción según el modelo de Endsley). La información proporcionada por esta fuente contiene los hechos concernientes a Eventos $Fa(Ev)$, umbrales $Fa(T_H)$ y los Indicadores de Rendimiento $Fa(KPI)$ del estado actual de la red.

- *Operador del Caso de Uso.* La base de conocimiento se obtiene a partir de la definición del caso de uso. El operador del caso de uso provee reglas de inferencia Ru (1), y declara objetos O (2), operaciones Op (3) y métricas de predicción Ft (4) que forman parte del proceso de análisis (ej. Qué observaciones se toman en cuenta (2), cómo (3), qué datos deben ser predichos (4) y cómo estos son considerados para adquirir conocimiento de un caso de uso específico (1)). Opcionalmente, el operador del caso de uso describe los umbrales adaptativos AT_h a ser calculados, y de ser el caso configura como el reconocimiento de patrones PR es aplicado.

- *Analizador.* Una parte importante de la información necesaria para un razonamiento apropiado es generada por el propio módulo de Análisis. La información es recolectada en grupos: percepción y lenguaje de Máquina. El primer bloque es imperativo y establece los hechos Fa a partir del reconocimiento de patrones $Fa(PR)$, la predicción $Fa(Ft)$ y los umbrales adaptativos $Fa(AT_h)$. Por otra parte, el lenguaje de máquina provee información adicional a la generada por el operador del caso de uso (definición de nuevas reglas Ru y descripción de las métricas de predicción Ft). Además, es posible generar información para mejorar la gestión del conocimiento (peso, priorización, fusión, suavizamiento, etc.).

- *Diagnóstico.* Las conclusiones finales y los síntomas que componen la Conciencia Situacional de SELFNET son enviadas al Módulo de Diagnóstico (Subcapa de Gestión Autónoma) [EN517] mediante reportes Re.

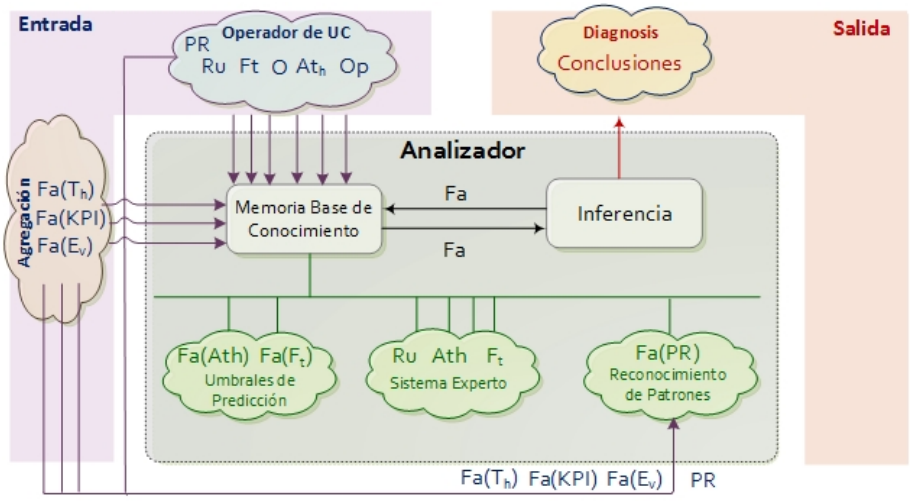

Figura 13.5: Módulo de análisis como caja negra

13.4 Descriptores de Caso de Uso

Esta sección describe las características de la información cuantitativa y sus categorías. En la tabla 13.1 se resume la información cuantitativa.

13.4.1 Objetos O

Los objetos (objects) $O = \{O_1, .., O_n\}$, $n \geq 1$ son los elementos a partir de los cuales el sistema infiere conocimiento, y que son añadidos a la base de conocimiento por el operador de caso de uso. Su función es describir la naturaleza de los datos para facilitar la selección del procesamiento adecuado y de los métodos de predicción. Los objetos son expresados de la siguiente forma:

$$O_i : \{object \quad name \mid weight \mid noValues \mid range \quad of \quad values \quad Va\}$$

Ejemplos:

$$\{Temperature \mid 1 \mid 1 \mid (-30°, 150°)\},$$
$$\{Link \quad Status \mid 0.7 \mid 1 \mid \{"Good", "Normal", "Bad"\}\},$$
$$\{Header \quad Encryption \mid 1.5 \mid 1 \mid (True, False)\},$$
$$\{Upper \quad Threshold \mid 2 \mid 1 \mid Y_t : t \epsilon T, \forall Y_i \epsilon R\}$$

Donde el nombre del objeto ($object \quad name$) actúa como identificador de la categoría del dato y el rango de valores ($rangeofvalues$) limita los valores que pueden ser asignados. El peso ($weight$) determina la prioridad y es un campo reservado para implementaciones futuras de aprendizaje automático. Finalmente, $noValues$ representa la cantidad de valores posibles. Debido a esto, un objeto puede ser especificado como una secuencia de k objetos definidos previamente o valores interrelacionados. En este caso, los objetos se representan como se indica a continuación.

$$O_i : \{object \quad name \mid weight \mid noValues \mid [Va_1][Va_2]...[Va_k]\}$$

Ejemplos:

$$\{pairWeather \mid 1 \mid 2 \mid [temperature][humidity]\},$$
$$\{metricA \mid 2 \mid 4 \mid [TTL][lenght][port][ipAddress]\},$$
$$\{tSerieB \mid 2 \mid 8 \mid [R][R][R][R][R][R][R][R]\},$$

La especificación de secuencias de un valor repetido varias veces en una fila se simplificad por el indicador : i, donde i es el número de veces que éste se repite. Por ejemplo, en el caso anterior:

$$\{tSerieB \mid 2 \mid 8 \mid [R][R][R][R][R][R][R][R]\},$$

Esta especificación puede ser simplificada del siguiente modo:

$$\{tSerieB \mid 2 \mid 8 \mid [R] : 8\},$$

13.4.2 Operaciones Op

Las operaciones (operations) $Op = \{Op_1, .., Op_n\}$, $n \geq 1$ representan las relaciones binarias entre hechos Fa, objetos O o sus valores posibles Va. Inicialmente, la base de conocimiento provee una batería básica de operaciones (Ej. Operaciones aritméticas, lógica proposicional, expresiones de estadística básica, etc.). Cuando un caso de uso es incorporado, el operador debe declarar el conjunto de operaciones y sus restricciones para el proceso de análisis, tal como se detalla a continuación:

$$Op_i : \{name \mid symbol \mid priority \mid operands \mid description\}$$

Ejemplos:

$$\{Equal \mid=\mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\},$$
$$\{LGT \mid\geq\mid 1 \mid (Fa, O, Va) \geq (Fa, O, Va) \mid left \quad is \quad GE\},$$
$$\{And \mid \wedge \mid 1 \mid (Fa) \wedge (Fa) \mid logical \quad conjuction\},$$
$$\{Addition \mid + \mid 3 \mid (Fa, Vo) + (Fa, Vo) \mid addition\},$$

Donde nombre (*name*) es el identificador de la operación de la batería, símbolo (*symbol*) es su nombre corto, prioridad (*priority*) representa su posición en la jerarquía de operaciones, operando (*operands*) limita las categorías de operandos aplicables en cada lado de la expresión binaria, y el campo descripción (*description*) explica brevemente su funcionalidad en lenguaje natural.

13.4.3 Hechos Fa

Los hechos (Facts) $Fa = \{Fa_1, .., Fa_n\}$, $n \geq 1$ son los elementos básicos del razonamiento de SELFNET. Estos son añadidos a la memoria del módulo de análisis por la capa de Agregación o deducidos por el motor de inferencia. Los hechos deben ser acompañados por una marca de tiempo (*timestamp*) que indica cuando han sido formulados, su ubicación (*location*) y un peso (*weight*) que determina su prioridad. La ubicación se refiere a los elementos de SELFNET (ej. máquinas físicas, nodos virtuales, etc.). El peso es un campo reservado para la prioridad del aprendizaje automático. La incertidumbre (*Uncertainty*) describe la posibilidad de que sea verdadero. Los hechos son descritos a través de la siguiente expresión:

$$Fa_i : \{expresion \mid weight \mid uncertainty \mid timestamp \mid location\}$$

Ejemplo:

$$\{Ur_{threshold} = MaxValue \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid NodeA\},$$
$$\{Temperature \geq 80° \mid 0.7 \mid 0.98 \mid Today \quad 03:41:20 \mid VM15\},$$
$$\{KPI7 = UrTh + MaxT \mid 1.2 \mid 1 \mid Today \quad 03:41:20 \mid VM15\},$$

13.4.4 Reglas Ru

Las reglas $rules = \{Ru_1, .., Ru_n\}$, $n \geq 1$ describen cómo el módulo de Análisis genera nuevo conocimiento mediante el sistema experto basado en reglas y utiliza una lógica proposicional, como por ejemplo:

$$(Fa(X) \vee Fa(Y)) \longrightarrow Fa(Z)$$
$$(Fa(X) \wedge Fa(Y)) \vee (Fa(X) \wedge \neg Fa(Z)) \longrightarrow Fa(B)$$
$$(Fa(C) \vee Fa(Y)) \vee \neg(Fa(A) \wedge Fa(Z)) \longrightarrow Fa(B)$$

Las reglas son acompañadas por un identificador de caso de uso (*usecase*) y su prioridad (*priority*) de inferencia. Es importante notar que para mejorar la escalabilidad, las reglas entre casos de uso son totalmente independientes entre sí. Las reglas se expresan del siguiente modo:

$$Rule : \{rule \mid priority \mid use \quad case\}$$

Ejemplos:

$$\{(Fa(X) \vee Fa(Y)) \longrightarrow Fa(Z) \mid 1 \mid SP\}$$
$$\{(Fa(B)) \longrightarrow Fa(Y) \mid 2 \mid SO\}$$
$$\{(Fa(X) \wedge Fa(Y)) \vee (Fa(X) \wedge \neg Fa(Z)) \longrightarrow Fa(B) \mid 1 \mid SH\}$$

13.4.5 Predicción Ft

La Predicción $Forecasts = \{Ft_1, .., Ft_n\}$, $n \geq 1$ representa la especificación de los objetos que deben ser estimados por cada caso de uso. De esta forma es posible mejorar la selección de los algoritmos y modelos de predicción. Dada la naturaleza de los ambientes de monitorización *a priori*, este enfoque considera solamente dos tipos de datos: series temporales y grafos. Las series temporales permiten estimar la evolución de los KPIs o de los umbrales en una ubicación concreta (infraestructura física, dispositivos de red virtualización, etc.). Por otra parte, los grafos facilitan la inferencia de cambios de la topología en regiones específicas de SELFNET, tales como congestión, inclusión de nuevos elementos o fallos de red. El sistema experto considera los resultados de predicción como hechos, por tanto Ft se refiere únicamente a la especificación cuando un nuevo caso de uso es incorporado. En la figura 13.4 se representan las predicciones como hechos ($Fa(Ft)$). A continuación se describe la expresión de predicción en series temporales:

$$Ft_i : \{timeSeries \mid object \mid domain \quad lenght\}$$

Ejemplos:

$$\{timeSeries \mid O_1 \mid obs \mid t + 5\}$$
$$\{timeSeries|O_2 \mid timestamp \mid Today13 : 28 : 15\}$$

Donde *timeSeries* es una palabra reservada que indica que la predicción es en series temporales, y dominio (*domain*) es la extensión de la predicción. Los ejemplos muestran dos palabras reservadas, *obs* (observaciones) and *timestamp* (marca de tiempo). Cuando

el tiempo es medido en observaciones (*obs*), la longitud de la predicción se indica a través del instante inicial de tiempo (t) y la cantidad de observaciones entrantes (ej. t+5 indica la predicción de las siguientes cinco observaciones). Por otra parte, *timestamp* indica que tan larga debe ser la predicción (Ej. Today 13:28:15 representa la predicción de cierto objeto entre el tiempo presente y las 13:28:15). Es importante notar que el término *timeSeries* es usado para describir la forma en la cual los datos son estructurados y no el algoritmo de predicción en sí mismo. La predicción puede ser realizada con métodos tradicionales de series temporales (suavizamiento exponencial, extrapolación, etc.) pero también con otras técnicas (redes neuronales, máquina de vector soporte, etc.). Es decisión del componente de predicción seleccionar la estrategia de predicción más adecuada. Si el proceso de predicción considera observaciones en grafos, la especificación es la siguiente:

$$Ft_i : \{graph \mid object \mid noVertex \mid domain \mid lenght\}$$

Ejemplos:

$$\{graph \mid O_1 \mid 30 \mid obs \mid t + 20\}$$
$$\{graph \mid O_2 \mid 45 \mid timestamp \mid Today19 : 12 : 07\}$$
$$\{graph \mid O_3 \mid 10 \mid timestamp \mid Today22 : 30 : 00\}$$

Donde grafo (*graph*) es la palabra reservada para declarar predicciones basadas en grafos. *object* representa la naturaleza de los datos en las aristas de su matriz de incidencia. $noVertex$ es el número de vértices (ej. dimensión $noVertex - by - noVertex$ de su matriz de adyacencia). Los dos últimos parámetros (*domain* y *length*) tienen la misma función de la expresión de predicción *timeSeries* (Indica la medida de tiempo y la extensión de la predicción).

13.4.6 Umbrales T_h

Los umbrales (thresholds) $T_h = \{T_{h1}, .., T_{hn}\}$, $n \geq 1$ representan el límite de tolerancia a fallos relacionados con valores asignados a objetos O. Estos son calculados por la tarea de Agregación de SELFNET, pero su especificación es parte del operador de caso de uso. Los umbrales son descritos mediate la siguiente expresión:

$$T_{hi} : T_h \quad name \mid object$$

Ejemplos:

$$\{maxTemp \mid O(temperature)\}$$
$$\{maxConnections \mid O(nConnections)\}$$
$$\{minQuality \mid O(QoS)\}$$

Donde T_h name es el identificador del umbral y *object* es el objeto sobre el cual es aplicado.

13.4.7 Umbrales Adaptativos T_h

Los umbrales adaptativos $AT_h = \{AT_{h1}, .., AT_{hn}\}, n \geq 1$ representan el límite de tolerancia a fallos relacionado con los valores asignados a las predicciones Ft. Estos son calculados por el componente de predicción del módulo de Análisis, pero debe ser especificados por el operador del caso de uso. De manera similar a los descriptores de predicción, AT_h toma en cuenta series temporales o grafos, como se describe a continuación:

$$AT_{hi} : AT_h \quad name \mid data \quad structure \mid CI \mid forecast$$

Ejemplos:

$$\{maxTemp \mid timeSeries \mid 0.95 \mid Ft(A)\}$$
$$\{maxWorkload \mid graph \mid 0.90 \mid Ft(X)\}$$

Donde AT_h $name$ es el identificador de los umbrales adaptativos, $data$ $structure$ es de tipo serie temporal ($timeSeries$) o grafos ($graph$), CI es el intervalo de confianza y forecast representa la predicción a partir del cual T_h es creado.

13.4.8 Reconocimiento de Patrones PR

La configuración de reconocimiento de patrones $PR = \{PR_1, ..., PR_n\}$, $n \geq 1$ describe cómo los hechos Fa relacionados con los datos de agregación son analizados para determinar su semejanza con la información de referencia establecida con anterioridad. Las salidas de las acciones del reconocimiento de patrones son hechos que muestran el grado de semejanza observado. Cada acción (PR) se representa como sigue:

$$PR_i : \{PR \quad name \mid objectIn \mid ObjectOut \mid action \mid reference \quad data\}$$

Ejemplos:

$$\{botnetTraffic \mid O(tFlow) \mid O(dist) \mid match \mid D(dataset1)\}$$
$$\{paylScan \mid O(payload) \mid O(dist) \mid anomaly \mid D(dataset2)\}$$
$$\{usrVerify \mid O(uAction) \mid O(dist) \mid anomaly \mid D(dataset3)\}$$

Donde PR $name$ es el identificador de la acción, $objectIn$ representa la naturaleza del dato a ser estudiado, $objectOut$ representa la naturaleza del objeto que recibe el grado de similitud, $action$ es la palabra reservada asociada con el tipo de análisis a ser ejecutado. Las acción por defecto es "match" para las observaciones coincidentes con los datos de referencia y "anomaly" para observaciones anormales. Finalmente, $referencedata$ representa el identificador de los conjuntos de datos (dataset) D a ser tomados en cuenta en el proceso de análisis.

13.4.9 Conjunto de Datos D

El conjunto de datos (Datasets) $D = \{D_1, ..., D_n\}$, $n \geq 1$ representa la información de referencia requeridos para llevar a cabo el reconocimiento de patrones. Dado que el módulo de Análisis no considera entrenamiento en línea, todos los datos de referencia son provistos por los casos de uso a través de la interfaz de usuario. El conjunto de datos es declarado por medio de la siguiente expresión:

$$D_i : \{D \quad name \mid object \mid type \mid source\}$$

Ejemplos

$$\{legitimatePayload \mid O(payload) \mid model \mid Repository1\}$$
$$\{mySet1|O(flowMetrics) \mid collection \mid Repository2\}$$
$$\{autoreplicationGens \mid O(binary) \mid signature \mid Repository3\}$$

Donde $D \quad name$ es el identificador del conjunto de datos y *object* representa la naturaleza de sus muestras. En este primer enfoque, el conjunto de datos puede ser de tres tipos: "collection", "model" o "signature". En primer lugar, "collection" hace referencia al conjunto de observaciones en bruto obtenidas directamente del entorno de monitorización. Por otra parte, "model" representa datos pre procesados. Finalmente, "signature" indica los patrones a ser identificados. El campo *source* determina donde se encuentra el conjunto de datos (ej: ruta, url, repositorio, etc.).

13.4.10 Conclusiones C

Las conclusiones $C = \{C_1, ..., C_n\}$, $n \geq 1$ son el subconjunto del grupo de hechos Fa que forman parte de la conciencia situacional de la red. Cuando una conclusión es inferida, esta es enviada al módulo de Diagnóstico [EN517] dado que es un indicador potencial de situaciones sospechosas en la red. Estos síntomas son definidos por el operador del caso de uso de la siguiente forma:

$$C_i : \{C \quad name \mid use \quad case \mid fact\}$$

Ejemplo:

$$\{gridlock \mid SP \mid Fa(A)\}$$
$$\{overHeating \mid SH \mid fa(X)\}$$

Donde C name es el identificador de la conclusión, use case representa el caso de uso definido en SELFNET, y fact es la conclusión desencadenada. Las conclusiones son reportadas al módulo de diagnóstico de la siguiente manera:

$$Re_i : \{C \quad name \mid use \quad case \mid fact \mid uncertainty \mid trigger\}$$

Ejemplo:

$$\{gridlock \mid SP \mid Fa(A) \mid 0.85 \mid Fa(B), Fa(C), Ru(1)\}$$
$$\{overHeating \mid SH \mid fa(X) \mid 0.75 \mid Fa(x), Ru(3)\}$$

Donde *uncertaintly* es la probabilidad de que la conclusión sea cierta, y trigger es la lista de reglas Ru o hechos Fa que fueron usados para su inferencia.

13.5 Ejemplos de Especificación y Flujos de Trabajos

Esta sección describe tres ejemplos de especificación de datos y los flujos de trabajo del Módulo de Análisis.

13.5.1 UC 1: Análisis de un Dispositivo de Temperatura

Esta sección describe un sensor relacionado con el caso de uso de auto reparación (Del inglés self-healing).

13.5.1.1 Descripción

El caso de uso ($myTemp$) requiere el identificador de síntomas relacionados con el recalentamiento de los dispositivos de red. Este es un ejemplo básico donde las capacidades de predicción y de umbrales adaptativos no son considerados. Por tanto los umbrales de decisión son estáticos y son construidos en la capa de agregación.

13.5.1.2 Estado Inicial

El módulo de Análisis dispone de una batería de operaciones predefinidas tales como cálculos de aritmética básica, funciones lógicas y estadísticas.

13.5.1.3 Especificación del Caso de Uso

Primeramente, el operador del caso de uso especifica los objetos básicos a ser tomados en cuenta: la temperatura de los dispositivos y su umbral superior.

$$O_1 : \{Temperature \mid 1 \mid 1 \mid R\}$$
$$T_{h1} : \{maxTemp \mid O_1\}$$

En segunda instancia, se indica los operadores que son requeridos y como estos son tomados en cuenta:

$$Op_1 : \{Equal \mid = \mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$$
$$Op_2 : \{LGT \mid \geq \mid 1 \mid (Fa, O, Va) \geq (Fa, O, Va) \mid leftisGE\}$$

Tercero, se definen las conclusiones:

$$C_1 : \{overheat \mid myTemp \mid Fa(O_1) \geq Fa(T_{h1})\}$$

El último paso es declarar las reglas de inferencia:

$$Ru_1 : \{Fa(O_1) \geq Fa(T_{h1}) \longrightarrow Fa(C_1) \mid 1 \mid myTemp\}$$

13.5.1.4 Flujo de Trabajo

En tiempo de ejecución, la capa de agregación notifica al módulo de Análisis los hechos (del inglés facts) relacionados con el caso de uso myTemp. Algunos de ellos afectan la temperatura de los dispositivos SELFNET, por ejemplo:

$$Fa_1 : \{O_1 = 35° \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid NodeA\}$$
$$Fa_2 : \{O_1 = 76° \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid NodeB\}$$
$$\ldots$$
$$Fa_5 : \{O_1 = 80° \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid NodeB\}$$

Es importante notar que la incertidumbre es 1 porque los sensores son deterministas (100 % de probabilidad de proveer la temperatura correcta). Los hechos toman en cuenta los umbrales estáticos:

$$Fa_7 : \{T_{h1} = 79° \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid All\}$$
$$Fa_8 : \{T_{h1} = 79° \mid 1 \mid 1 \mid Today \quad 12:22:16 \mid All\}$$

Los hechos son provistos por la capa de agregación y son directamente incluidos en la memoria del módulo de Análisis. Si los hechos son actualizados para la misma ubicación (ej. Fa_5 and Fa_6), la última versión es considerada por el motor de inferencia. Después de cierto período de observación, el motor de inferencia intenta deducir nuevo conocimiento a partir del conjunto de reglas de cada caso de uso. En myTemp, el módulo de Análisis trata de inferir conclusiones para la regla Ru_1. En el tiempo $Today \quad 12:22:17$ el sistema satisface la primera conclusión: $Fa_5(O_1 = 80°) \geq Fa_8(O_1 = 79°)$, por tanto el hecho $Fa(C_1)$ es añadido a la memoria:

$$Fa_9 : \{Fa_5 \geq Fa_8 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid NodeB\}$$

La ubicación del nodo NodeB es considerada debido a que es la más restrictiva entre NodeB,All. Por tanto se infiere el síntoma C_1 que luego será reportado al Módulo de Diagnóstico:

$$Re_1 : \{overheat \mid myTemp \mid Fa_9 \mid 1 \mid Fa_5, Fa_8, Ru_1\}$$

Finalmente, el motor de inferencia continuará buscando más síntomas.

13.5.2 UC 2: Análisis de la Congestión de Red

En esta sección se describe un ejemplo de un sensor para el caso de uso de auto optimización (del inglés self-optimization).

13.5.2.1 Descripción

El caso de uso "Auto congestión" (Del inglés SC) requiere el identificador de síntomas relacionados con la congestión de tráfico de los elementos de SELFNET. En este ejemplo; la predicción y los umbrales adaptativos son considerados.

13.5.2.2 Estado Inicial

El módulo de Análisis dispone de una batería de operaciones predefinidas tales como cálculos de aritmética básica, funciones lógicas y estadísticas.

13.5.2.3 Especificación del Caso de Uso

Primeramente, el operador de caso de uso especifica los objetos básicos a ser tomados en cuenta: el nivel de congestión y su predicción.

$$O_1 : \{congestion \mid 1 \mid 1 \mid [0,1]\}$$
$$Ft_1 : \{timeSeries \mid O_1 \mid obs \mid t+3\}$$

Luego, se define el umbral adaptativo a ser generado automáticamente a partir de la información provista por el historial de seguimiento y el módulo umbral adaptativo (*AdaptiveThresholding*).

$$AT_{h1} : \{maxCongestion \mid timeSeries \mid 0.95 \mid Ft_1\}$$

En segundo lugar, se especifica que operadores son requeridos y cómo estos son tomados en cuenta:

$$Op_1 : \{Equal \mid=\mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$$
$$Op_2 : \{LGT \mid\geq\mid 1 \mid (Fa, O, Va) \geq (Fa, O, Va) \mid leftisGE\}$$

Tercero, las conclusiones son identificadas:

$$C_1 : \{gridlock \mid SC \mid Fa(O_1) \geq Fa(AT_{h1})\}$$

El último paso consiste en declarar las reglas de inferencia:

$$Ru_1 : \{Fa(O_1) \geq Fa(AT_{h1}) \longrightarrow Fa(C_1) \mid 1 \mid SC\}$$

13.5.2.4 Flujo de Trabajo

En tiempo de ejecución, la capa de agregación notifica al módulo de Análisis los hechos relacionados con el caso de uso SC, por ejemplo:

$$\begin{aligned}
Fa_1 &: \{O_1 = 0.6 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ServerA\}\\
Fa_2 &: \{O_1 = 0.65 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ServerA\}\\
Fa_3 &: \{O_1 = 0.65 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ServerA\}\\
Fa_4 &: \{O_1 = 0.67 \mid 1 \mid 1 \mid Today \quad 12:22:21 \mid ServerA\}\\
Fa_5 &: \{O_1 = 0.68 \mid 1 \mid 1 \mid Today \quad 12:22:24 \mid ServerA\}\\
&\ldots\\
Fa_{44} &: \{O_1 = 0.66 \mid 1 \mid 1 \mid Today \quad 12:22:50 \mid ServerA\}\\
Fa_{45} &: \{O_1 = 0.67 \mid 1 \mid 1 \mid Today \quad 12:22:52 \mid ServerA\}\\
Fa_{47} &: \{O_1 = 0.69 \mid 1 \mid 1 \mid Today \quad 12:22:56 \mid ServerA\}\\
Fa_{48} &: \{O_1 = 0.86 \mid 1 \mid 1 \mid Today \quad 12:22:58 \mid ServerA\}\\
Fa_{49} &: \{O_1 = 0.97 \mid 1 \mid 1 \mid Today \quad 12:23:01 \mid ServerA\}
\end{aligned}$$

La construcción de un modelo predictivo requiere una cierta cantidad de observaciones previas; en este caso, se ha considerado los primeros 45 hechos (facts). Estos son manipulados por el historial de seguimiento para extraer la información necesaria y definir la serie temporal. En el tiempo Today 12:22:52 (cuando el hecho Fa_{45} es deducido), el componente de predicción provee la primera predicción Ft_1 para cada instancia t+3. Luego un nuevo hecho es incluido en la memoria:

$$Fa_{46} : \{AT_{h1} = 90 \mid 1 \mid 1 \mid Today12:23:52 \mid ServerA\}$$

Este hecho desencadena la regla Ru_1, porque $Fa_{49}(O_1 = 0.97) \geq Fa_{46}(AT_{h1} = 90)$, y la conclusión C se cumple. El nuevo conocimiento $Fa(C_1)$ es añadido a la memoria como:

$$Fa_{50} : \{Fa_{49} \geq Fa_{46} \mid 1 \mid 1 \mid Today12:23:01 \mid NodeB\}$$

Finalmente, el síntoma es reportado al módulo de diagnóstico de la siguiente forma:

$$Re_1 : \{gridlock \mid SC \mid Fa_{50} \mid 1 \mid Fa_{49}, Fa_{46}, Ru_1\}$$

13.5.3 UC 3: Análisis de la Carga Util

Esta sección describe un sensor relacionado con el caso de auto protección (del inglés self-protection).

13.5.3.1 Descripción

El caso de uso "Auto Protección" (del inglés SG) indentifica síntomas relacionados con carga útil anómala en el tráfico de SELFNET. En este ejemplo, se considera el reconocimiento de patrones.

13.5.3.2 Estado Inicial

El módulo de Análisis dispone de una batería de operaciones predefinidas,tales como cálculos de aritmética básica, funciones lógicas y estadísticas. Los repositores externos (Rep1, Rep2) proveen una colección legítima (Rep1) y una maliciosa (Rep2) de las observaciones del tráfico de SELFNET.

13.5.3.3 Especificación de Caso de Uso

Primero, el operador de caso de uso especifica los objetos básicos a ser tomados en cuenta; en este ejemplo hay una carga útil del tráfico de SELFNET O_1, su similitud con el conjunto legítimo de la carga útil O_2 de las muestras maliciosas O_3.

$$O_1 : \{payload \mid 1 \mid 1 \mid hexadecimal\}$$
$$O_2 : \{simLegi \mid 1 \mid 1 \mid \{0..1\}\}$$
$$O_3 : \{simMal \mid 1 \mid 1 \mid \{0..1\}\}$$

En segundo lugar, se especifica que operaciones son requeridas y como son tomadas en cuenta:

$$Op_1 : \{Equal \mid =\mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$$
$$Op_2 : \{LT \mid >\mid 1 \mid (Fa, O, Va) > (Fa, O, Va) \mid leftisG\}$$

Luego, se define el conjunto de datos:

$$D_{legi} : \{legitimatePayload \mid O(payload) \mid collection \mid Rep1\}$$
$$D_{mal} : \{maliciousPayload \mid O(payload) \mid collection \mid Rep2\}$$

Y luego las acciones de reconocimiento de patrones son ejecutadas:

$$PR_1 : \{legMeasure \mid O_1 \mid O_2 \mid anomaly \mid D(D_{legi})\}$$
$$PR_2 : \{malMeasure \mid O_1 \mid O_3 \mid anomaly \mid D(D_{mal})\}$$

Las conclusiones son identificadas de la siguiente forma:

$$C_1 : \{maliciousContent \mid SC \mid Fa(O_2) < Fa(O_3)\}$$

Y las siguientes reglas son incorporadas:

$$Ru_1 : \{Fa(O_2) < Fa(O_3) \longrightarrow Fa(C_1) \mid 1 \mid SP\}$$

13.5.3.4 Flujo de Trabajo

En tiempo de ejecución, la capa de agregación notifica al módulo de Análisis los hechos relacionados con el caso de uso SG:

$$\begin{gathered}
Fa_1 : \{O_1 = FF217 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionA\} \\
Fa_2 : \{O_1 = FFFFF \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionB\} \\
Fa_3 : \{O_1 = 00DE8 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionA\} \\
Fa_4 : \{O_1 = A4FC9 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionA\} \\
Fa_5 : \{O_1 = FF218 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ConexionC\} \\
\ldots \\
Fa_{38} : O_1 = F0279 \mid 1 \mid 1 \mid Today \quad 12:22:23 \mid ConexionA
\end{gathered}$$

Cada vez que la carga útil es observada, el módulo de Análisis de SELFNET ejecuta las acciones de reconocimiento de patrones PR_1 y PR_2, retornando los siguientes hechos:

$$\begin{gathered}
Fa_{1R1} : \{O_2 = 0.9 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionA\} \\
Fa_{1R2} : \{O_3 = 0.2 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionA\} \\
Fa_{2R1} : \{O_2 = 0.9 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionB\} \\
Fa_{2R2} : \{O_3 = 0.18 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionB\} \\
Fa_{3R1} : \{O_2 = 0.8 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ConexionC\} \\
Fa_{3R2} : \{O_3 = 0.21 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ConexionC\} \\
\ldots
\end{gathered}$$

En el tiempo Today 12:22:23 se identifican los siguientes hechos:

$$\begin{gathered}
Fa_{32R1} : \{O_2 = 0.66 \mid 1 \mid 1 \mid Today \quad 12:22:23 \mid ConexionA\} \\
Fa_{32R2} : \{O_3 = 0.92 \mid 1 \mid 1 \mid Today \quad 12:22:23 \mid ConexionA\}
\end{gathered}$$

Se desencadena la regla Ru_1, debido a $Fa_{32R1}(O_2 = 0.66) < Fa_{32R2}(O_3 = 0.92)$, y luego las conclusión C es satisfecha. El nuevo conocimiento $Fa(C_1)$ es añadido a la memoria como:

$$Fa_{50} : \{Fa_{32R1} < Fa_{32R1} \mid 1 \mid 1 \mid Today \quad 12:23:23 \mid ConexionA\}$$

Finalmente, el síntoma es reportado al módulo de Diagnóstico:

$$Re_1 : \{suspiciousPayload \mid SC \mid Fa_{50} \mid 1 \mid Fa_{32R1}, Fa_{32R2}, Ru_1\}$$

Cuadro 13.1: Resumen de la especificación de caso de uso

Dato	**Categoría**	**Proveedor**	**Destino**	**Formato**
Objecto (simple) O	Especificación	Caso de Uso	Analizador	$O_i : \{object \quad name \mid weight \mid noValues \mid range \quad of \quad values \quad Va\}$
Objecto (mult) O	Especificación	Caso de Uso	Analizador	$O_i : \{object \quad name \mid weight \mid noValues \mid [Va_1][Va_2]...[Va_k]\}$
Operación Op	Especificación	Caso de Uso	Analizador	$Op_i : \{name \mid symbol \mid priority \mid operands \mid description\}$
Hechos Fa	Evaluación	Agregación Análisis	Analizador	$Fa_i : \{expresion \mid weight \mid uncertainty \mid timestamp \mid location\}$
Reglas Ru	Especificación	Caso de Uso	Analizador	$Rule : \{rule \mid priority \mid use \quad case\}$
Predicción (ts) Ft	Especificación	Caso de Uso	Analizador	$Ft_i : \{timeSeries \mid object \mid domain \quad lenght\}$
Predicción (G) Ft	Especificación	Caso de Uso	Analizador	$Ft_i : \{graph \mid object \mid noVertex \mid domain \mid lenght\}$
Umbrales T_h	Especificación	Caso de Uso	Analizador	$T_{hi} : \{T_h \quad name \mid object\}$
A. Umbrales At_h	Especificación	Caso de Uso	Analizador	$AT_{hi} : \{AT_h \quad name \mid data \quad structure \mid CI \mid forecast\}$
Conjunto de Datos D	Especificación	Caso de Uso	Analizador	$D_i : \{D \quad name \mid object \mid type \mid source\}$
Recono. de Patrones PR	Especificación	Caso de Uso	Analizador	$PR_i : \{PR \quad name \mid objectIn \mid ObjectOut \mid action \mid reference \quad data\}$
Conclusión C	Especificación	Caso de Uso	Analizador	$C_i : \{C \quad name \mid use \quad case \mid fact\}$
Reporte Re	Reporte	Analizador	Diagnóstico	$Re_i : \{C \quad name \mid use \quad case \mid fact \mid uncertainty \mid trigger\}$

13.6 Resumen

En este capítulo, se explica la aplicación de las capacidades de análisis e inteligencia en redes 5G. Se introduce el diseño del Módulo de Análisis de SELFNET y su especificación de datos. El diseño permite el reconcimiento de patrones, el razonamiento y la predicción para inferir situaciones anormales en el comportamiento de la red, facilitando de esta forma las tareas de diagnóstico y de toma de decisiones. La principal contribución del módulo de Análisis de SELFNET es que provee un enfoque general, simple y escalable, que permite nuevas reglas y métricas en el proceso de análisis cuando un nuevo caso de uso es añadido por el operador. Los sensores de SELFNET recolectan información de diferentes fuentes de datos tales como elementos virtuales, LTE, SDN y dispositivos tradicionales de red; en consecuencia la información recolectada puede ser sujeta a un proceso de análisis. Además, la propuesta fue elaborada de tal forma que permite añadir nuevas capacidades de análisis a través de un enfoque basado en plugins.

Capítulo 14

Conclusiones y Trabajos Futuros

En términos generales, las capacidades de análisis e inteligencia juegan un rol importante para cumplir los requerimientos de las redes 5G, en combinación con tecnologías emergentes tales como SDN, NFV o la computación en la nube. Todos estos campos pueden tomar ventaja de los conceptos de predicción, reconocimiento de patrones o inteligencia artificial. De esta forma, las redes 5G esperan proveer capacidades avanzadas relacionadas con la gestión de la red y la detección de problemas o fallos de red. Por su parte, el diagnóstico de la información es requerida para conocer cuál es la causa real de un evento. Debido a ello, la inteligencia de red es provista en dos fases: i) fase de análisis y ii) fase de toma de decisiones; de manera similar a una evaluación médica donde en primera instancia los síntomas son detectados y en base a éstos un tratamiento es aplicado.

Los resultados de la investigación presentada en esta tesis ayudan a cubrir las necesidades de inteligencia de las redes 5G a través de la introducción del análisis de datos y el concepto de conciencia situacional. Esta propuesta toma en cuenta SDN y NFV aplicados a este tipo de sistemas. En primer lugar, este trabajo propone una arquitectura general para la gestión de incidencias basado en la combinación del modelo de conciencia situacional propuesto por Endsley con las directrices de gestión de riesgo y estándares tradicionales. De esta manera, se facilita la automatización del despliegue de contramedidas tanto forma proactiva como reactiva. La arquitectura propuesta toma también ventaja de cada fuente de información de 5G, con miras a considerar la calidad del contexto en el proceso de toma de decisiones.

Como se indicó a lo largo del documento, la presente tesis es parte del proyecto SELFNET el cual provee un marco para la gestión autónoma e inteligente para redes móviles 5G. SELFNET posibilita el despliegue automático de funciones SDN/NFV y la reconfiguración de los dispositivos de red y los parámetros de los servicios con la finalidad de mitigar problemas existentes o futuros, sin afectar la calidad de servicio del usuario final o infringiendo los SLAs. Uno de los principales retos de SELFNET es analizar la información de cinco fuentes heterogéneas y proveer un enfoque escalable y basado en casos de uso capaz de brindar soporte para diferentes tipos de reglas y funcionalidades de análisis. En este contexto, este trabajo propone el módulo de Análisis de SELFNET.

El Marco de Análisis de SELFNET provee un esquema de propósito general fácilmente adaptable a las necesidades del operador y por lo tanto capaz de superar las restricciones

de diseño en diferentes entornos de monitorización debido a la gran cantidad de tecnologías que son parte de los escenarios 5G. El Marco de Análisis de SELFNET es capaz de analizar la información de diferentes fuentes heterogéneas tales como elementos SDN, dispositivos virtuales o métricas de sensores especializados. Otra característica importante es que el módulo de Análisis facilita la incorporación de diferentes estrategias de análisis, tal como algoritmos innovadores de predicción o reconocimiento de patrones. Este marco es capaz de operar indistintamente con diferentes paradigmas de minería de datos o lenguaje de máquina, entre ellos big data, información convencional, etc. El uso de cualquiera de estos métodos no implica cambios en el diseño, siendo simplemente un problema de implementación. Como resultado, esta propuesta es fácilmente adaptable a futuras tecnologías y enfoques.

La especificación de datos propuesta para la incorporación de nuevos casos de uso es simple y ajustable. SELFNET implementa una triada de servicios auto protección, auto recuperación y auto optimización con dependencias y características completamente diferentes (métricas, dispositivos de red a ser monitorizados, algoritmos de predicción y reconocimiento de patrones, etc.). Es importante enfatizar que la incorporación de nuevos casos de uso está basado solamente en cambios de configuración, sin la necesidad de modificar la implementación del módulo de Análisis o incluir software adicional. Sin embargo, la efectividad del Marco de Análisis de SELFNET depende de la calidad de la especificación definida por el operador cuando configura las funcionalidades del Analizador.

En resumen, El módulo de Análisis de SELFNET provee reconocimiento de patrones, capacidades de razonamiento y predicción para inferir síntomas frente a situaciones específicas, facilitando las tareas de diagnóstico y de toma de decisiones. La principal contribución del módulo de Análisis de SELFNET es su enfoque general, simple y escalable, permitiendo nuevas reglas y métricas en el proceso de análisis cuando un nuevo caso de uso es añadido por el operador. Además, esta propuesta fue construida para dar soporte nuevas capacidades de análisis a través de un enfoque basados en plugins y tomando en cuenta el concepto de conciencia situacional. Este trabajo también propone la especificación de datos para definir los datos de entrada a ser considerados en el proceso de diagnóstico.

14.1 Trabajos Futuros

Teniendo en cuenta que la introducción de inteligencia y el análisis de datos en redes 5G están en una fase temprana de desarrollo y por otra parte las consideraciones iniciales del Marco de Análisis de SELFNET, los siguientes trabajos futuros pueden ser llevados a cabo.

- Primeramente, el Marco de Análisis de SELFNET considera solamente la fase de diagnóstico para facilitar el proceso de toma de decisiones, el cual es considerado parte del trabajo futuro. Basado en los síntomas provistos por el módulo de Análisis, la fase de toma de decisiones lleva a cabo tanto respuestas activas como proactivas, cerrando así el ciclo de inteligencia requerido por SELFNET.

- Esta propuesta no toma en cuenta un sistema de seguimiento de contramedidas

o retroalimentación de la fase de toma de decisiones. Por tanto, este aspecto será cubierto en trabajos futuros.

- Esta propuesta es construida sobre la suposición que las observaciones normales a ser analizadas coinciden con las muestras de referencia del proceso de aprendizaje. El entorno estacionario propuesto por el módulo de Análisis de SELFNET proporciona una solución simple y eficiente, pero es propenso a pequeños fallos cuando ocurren cambios. Por tanto, la introducción de mecanismos para trabajar en entornos de monitorización no estacionario es también parte del trabajo futuro.

- El módulo de Análisis de SELFNET no es capaz de lidiar con ambientes de monitorización complejos, donde la calidad del proceso de análisis decrece con el tiempo. Dado la importancia de este tipo de escenarios, serán estudiados en el futuro.

- SELFNET es un escenario de monitorización complejo donde una gran cantidad de sensores recolectan información procedente de diferentes fuentes de datos en tiempo real. La información (agregada o correlacionada) debe ser periódicamente empaquetada, cargada y convertida a hechos, para que en lo posterior el módulo de Análisis lleve a cabo cálculos complejos sobre los mismos. En este contexto, otro aspecto a estudiar es la identificación de indicadores cualitativos relacionados con la granularidad de la información contenida en un "paquete agregado". Este punto en particular permitirá mejorar la efectividad de las acciones de análisis.

- Otro aspecto importante a tener en mente es el orden de ejecución de los componentes de Análisis para proveer consistencia y facilitar la organización de la información recibida. Por tanto, la investigación de métodos para procesar y analizar la información recibida es parte también del trabajo en curso.

Part IV

Papers Related to This Thesis

Appendix A

List of Papers

1. L. I. Barona López, A. L. Valdivieso Caraguay, L. J. García Villalba: Extending OpenFlow in Virtual Networks. In The 7th International Conference on Information Technology, Amman, Jordan, May 12 – 15, 2015.

2. L. I. Barona López, A. L. Valdivieso Caraguay, L. J. García Villalba, D. López: Trends on Virtualisation with Software Defined Networking and Network Function Virtualisation. IET Networks, 4(5): 255–263, September 2015.

3. L. I. Barona López, J. Maestre Vidal, A. L. Valdivieso Caraguay, M. A. Sotelo Monge, L. J. García Villalba, et al: Seguridad en Redes de Nueva Generación: Arquitectura para la Gestión de Incidencias. Actas de la XIV Reunión Española sobre Criptografía y Seguridad de la Información (RECSI 2016), pages 1–6, Menorca-Islas Baleares, España, October 26–28, 2016.

4. L. I. Barona López, A. L. Valdivieso Caraguay, M. A. Sotelo Monge, L. J. García Villalba, et al: Key Technologies in the Context of Future Networks: Operational and Management Requirements. Future Internet, 9(1): 1–15, December 2016.

5. L. I. Barona López, J. Maestre Vidal, A. L. Valdivieso Caraguay, M. A. Sotelo Monge, L. J. García Villalba, et al: Towards Incidence Management in 5G based on Situational Awareness. Future Internet 2017, 9(3): 1-15, January 2017.

6. L. I. Barona López, J. Maestre Vidal, L. J. García Villalba: An Approach to Data Analysis in 5G Networks. In MDPI Entropy, 19(2): 1–23, February 2017.

7. L. I. Barona López, J. Maestre Vidal, L. J. García Villalba: Orchestration of Use Case driven Analytics in 5G Scenarios. In Journal of Ambient Intelligence and Humanized Computing. (Accepted April 2017).

Additional Publications

1. J. P. Santos, R. Alheiro, L. I. Barona López, L. J. García Villalba, et al: SELFNET Framework Self-healing Capabilities for 5G Mobile Networks. Transactions on Emerging Telecommunications Technologies, 27(9): 1225-1232, June 2016.

2. L. J. García Villalba, L. I. Barona López, A. L. Valdivieso Caraguay. Use Cases Definition and Requirements of the System and its Components. SELFNET Project, October 2015. $https://doi.org/10.18153/SLF-671672-D2\_1$.

3. L. J. García Villalba, L. I. Barona López, A. L. Valdivieso Caraguay. Report and Prototypical Implementation of the Monitoring and Discovery Module. SELFNET Project, September 2016.

The 7th International Conference on Information Technology

Big Data

ISSN 2306-6105

ISBN 978-9957-8583-3-9

Conference Proceeding

Full

Prepared and Edited by:

- ICIT15 General Chair
  - Ali Al-Dahoud, Dean of Science and IT Faculty
- Editorial Board
  - Hani Mimi, ICIT15 Co-chair
  - Khalid Jaber, ICIT5 Co-chair
  - Israa Sabatin, Designer
  - Hanade Al-Shawabkeh, Editor
  - Ayman Al-Qafa'an, Editor

The Hashemite Kingdom of Jordan

The Deposit Number at the National Library

(2015/4/1450)

009

نسخة / مركز الايداع

Information Technology (7:Amman:2015)

The 7th International Conference on Information Technology / Ali As'ad Al-Dahoud. – Amman: Al-Zaytoonah University of Jordan, 2015

(163) p.

Deposit No. : 2015/4/1450

Descriptions: /Computers //Conferences//Information Technology/

ICIT 2015 The 7th International Conference on Information Technology
doi:10.15849/icit.2015.0037

Extending OpenFlow in Virtual Networks

Lorena Isabel Barona López, Ángel Leonardo Valdivieso Caraguay, Luis Javier García Villalba
Group of Analysis, Security and Systems (GASS)
Department of Software Engineering and Artificial Intelligence (DISIA)
Faculty of Information Technology and Computer Science, Office 431
Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9
Ciudad Universitaria, 28040 Madrid, Spain
Email: {lorebaro, angevald}@ucm.es, javiergv@fdi.ucm.es

***Abstract*— Software Defined Networking (SDN) is a novel technology that has become a prominent topic in the last years. In any research is essential to have emulators and simulators in order to test new applications or protocols. In this context, we present the integration of OpenFlow protocol with Virtual Networks over linuX (VNX) tool, as new alternative for the emulation with SDN. VNX/OpenFlow approach integrates three kind of tools, an OpenFlow compliant switch (Open vSwitch), Network Operative Systems (POX, NOX and Beacon) and finally tools to control the performance and the network traffic. For the validation process, we present two VNX/OpenFlow scenarios to test the correctness of this tool. Finally, the result of this work allows the deployment of virtual scenarios with OpenFlow protocol.**

Keywords*— Emulation, OpenFlow, Software Defined Networking; Virtualization.

I. INTRODUCTION

Network data traffic has grown exponentially in the last years due the emergence of real time applications, video streaming, the rise of social networking, the introduction of cloud computing, among others. The research community has created protocols in order to cover these new needs, however the standardization process takes a long time and the improvements in communication methods and information processing are almost nonexistent [1].

Existing networks should have an open control and provide a real environment to tests with production traffic, due to these requirements the concept of Software Defined Networking arises [2]. SDN is not a new concept, rather is the result of many research projects such as the Active Networks and Ethane project [3]. SDN takes advantage of the best characteristics of these technologies (programmability, control and data plane separation), changing the way we see networks today. SDN allows the separation of data and control plane in network devices [4]. The control of the network behavior is in charge of an external device known as Network Operative System (NOS). The communication between network devices and the controller is established with a defined protocol, the most known OpenFlow [5].

Currently, a great number of enterprises like Google have incorporated OpenFlow in their infrastructures and devices, and there are some organizations, such as Open Networking Foundation (ONF), which promote the development and the widespread of OpenFlow and SDN architecture. There are few projects to test with SDN such as simulators, emulators or testbeds. One of the first OpenFlow testbed was developed by Global Environment for Network Innovations (GENI) [6], which interconnects the principal universities of United States. Likewise, the project OpenFlow in Europe: Linking Infrastructure (OFELIA) [7] connects 8 OpenFlow islands, allowing experimentation with this technology.

Other interesting tool is ns-3 simulator [8]. Although ns-3 has support for OpenFlow, it does not work with typical controllers such as POX [9], NOX [10], Beacon [11], Floodlight [12], OpenDaylight, and so on. Instead, ns-3 has its own OpenFlow controller. Regarding OpenFlow emulators, the most known is Mininet which is used for rapidly prototyping large networks [13]. Mininet can run real applications with a great variety of topologies; however, the performance fidelity depends on the CPU capacity and the number of the emulated hosts. Additionally, there is a hybrid approach that combines simulation and emulation in one tool called EstiNet [14]. It has not problems with fidelity performance, however, it is not a free tool.

There is a wide range of tools for experimentation with virtual networks, such as the virtualization tool called Virtual Networks over linuX (VNX) [15]. VNX is used in education and research fields, for instance in the experimentation with Intrusion Detection Systems (IDS), Multipath TCP (MTCP), among others. This paper presents the integration of this tool with OpenFlow protocol. For this purpose, OpenFlow-enabled switch and controllers are integrated.

This work has been divided into five sections, as follows: The second section contains an introduction of SDN and OpenFlow protocol. Then, the third section presents the description of simulation and emulation tools. Next, the fourth

doi:10.15849/icit.2015.0037

section shows the VNX-OpenFlow integration process and the validation of two test scenarios. Finally, a discussion is opened in the fifth section.

II. SOFTWARE DEFINED NETWORKING

Software Defined Networking introduces a paradigm change in the network communication, facilitating the innovation and the network programmability. SDN proposes the separation between the control and the data plane in networking devices. Consequently, the network is more flexible, programmable and it has automation capabilities. The own device could carry out advanced capacities such as firewall rules, load balancing, among others.

The control of whole network is performed by a central point known as a controller. The network devices are connected with the controller through secure communication channel like Sockets Secure Layer (SSL). In the communication process is needed a standardized protocol the most known OpenFlow [5], which defines the communication rules between controller and OpenFlow compliant switches. OpenFlow offers new features that enable experimentation without expose the internal structure of switches from different vendors. For this purpose, OpenFlow delimits the basic functions of OpenFlow switches based on common characteristics of traditional Ethernet switch. OpenFlow defines three kind of tables, these are: flow, group and meter table. OpenFlow also introduces the flow concept, which can be defined as a kind of traffic such as the http requests, traffic to the same destination address, and so on. Moreover, OpenFlow establishes a pipeline in order to process the incoming packets. The packet is first matched against flow entries of flow table 0 and may continue with the next tables, depends on the result of the match in the table. Flow entries match packets based on the priority field (highest priority). If a flow entry is found, the instructions are executed (Modify packet and update match fields, update action set, update metadata). If the packet does not match with a flow entry in any table, the outcome depends on the configuration of the table miss. A possible action is to search in the next table. Based on the SDN architecture and the business requirements many tools have been developed, such as:

- Virtualization tools [16].
- Network Operating System (controllers) [9] [10].
- Virtual switches [17].
- Tools for Quality of Services and Quality of Experience [18].
- Management [19] [20].
- Optical Networks [21] [22].
- Traffic engineering and load balancing [23].
- Load Balancing [24].
- Simulation and Emulation tools [8] [13] [14].

All of these research fields are deployed and tested through some approaches; real testbeds, emulator or simulators [25]. OpenFlow testbeds [6] [7] allow the experimentation in real environments on a large scale. However, testbeds are not easily accessible by potential researchers. For its part, simulation and emulation approaches provide facilities in terms of scalability, portability and accessibility in the case of open source tools. Nevertheless, in some cases they produce inaccurate outcomes. We describe some familiar tools ns-3, Mininet and EstiNet, as well as VNX/OpenFlow.

III. SIMULATION AND EMULATION TOOLS

NS-3 is a simulator tool focuses on research and educational fields. It is an open sources simulator that provides an extensible network platform with several external animators, data analysis and visualization tools. In order to enable the simulation with OpenFlow protocol, Ns-3 implements its OpenFlow-enabled switch and its own controller, as a modules written in C++. The switch component is known as *OpenFlowSwitchNetDevice*. This object consists of a set of net devices that represent the switch ports, according to the OpenFlow Switch Specification v0.8.9. Even though Ns-3 can be used for real-time simulations, there are some issues that the user should take into account such as the slow learning curve to use the tool, the compatibility with a basic OpenFlow version (0.89) and specially it does not run a typical OpenFlow controller. Therefore, the controller applications generated with ns-3 controller cannot be used in real network. If a controller like Pox or Floodlight was required, these will need substantial modifications.

For its part, Lantz et al. in [13] proposes Mininet, a virtualization tool for rapidly prototyping large networks in a single laptop. This tool includes OpenFlow support and combines lightweight virtualization capabilities over Linux operative system with an extensible CLI and API. A scenario built with Mininet is deployable, interactive, scalable, realistic and it can easily share. In fact, the Mininet topologies and the controller applications can be used for others researchers without modifications in the emulation environment as well as in real networks. Mininet run on virtual machine monitors like VMWare, XEN and VirtualBox or it can be installed in a Linux system. Mininet allows the deployment of hundreds of nodes, emulating OpenFlow-enabled switches, controllers like POX, virtual links and hosts. Mininet shares components like the file system, the user ID space, the kernel, device drivers, among others. Tough, Mininet is the most popular tool for SDN has limitations of performance fidelity related with the available resources, real bandwidth and the timing of the process.

A novel hybrid approach has recently presented called EstiNet [14]. This combines the best characteristics of both simulation and emulation mode in one tool. On the one hand, it allows the deployment of large networks in a flexible, easy, scalable and repeatable way. On the other hand, EstiNet takes into account the timing needs for real applications in order to obtain the same results in both, virtual and real deployments. EstiNet supports 1.3.2 OpenFlow Switch Specification and it can run NOX, POX, Floodlight, and Ryu controllers without any modifications. For this purpose, EstiNet intercepts the packets between two real applications through tunnel network interfaces and redirects the packets to the EstiNet simulation engine. The entire process is based on a simulation clock, which allows accurate results. Besides, EstiNet provides a graphical user interface for configure the scenarios and observe

doi:10.15849/icit.2015.0037

the outcomes from the simulations. The results of this tool show better scalability and performance than Mininet, however their main problem is that it need a payment for the tool. The universities can embrace the EstiNet University Program. This grants a license during six months with a cost of US$1500 or a license to 12 months for US$2500, becoming its main disadvantage.

As we have seen, there are few tools or testbeds that allow the SDN experimentation. We present VNX a modular architecture based on plugins (Fig. 1), which allows the deployment of virtual testbeds. This tool includes the code of the previous tool VNUML [26].

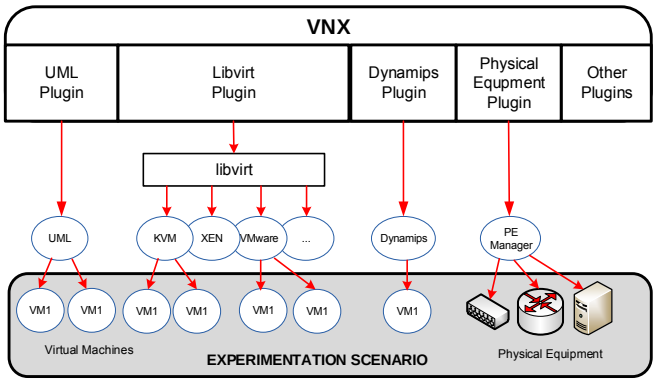

Fig. 1. VNX Architecture [15].

The plugins used by VNX are:

- UML (User Mode Linux) can be considered a hypervisor-based technique.
- libvirt allows virtualization capabilities and some virtualization platforms, such as Xen, VMware, KVM, VirtualBox, etc.
- Dynamips plugin allows the emulation the hardware of Cisco routers.
- Olive allows the integration of Juniper routers.
- Physical equipment plugin, which allows the connection between VNX physical islands.

VNX is a free tool based on Linux that allows the easy creation and management of large virtual scenarios over a single server or a cluster. The scenarios can have nodes in some physical hosts and can use different operative systems, for example Linux and Windows. In turn, each physical host can deploy their own virtual testbed. Besides, VNX allows the creation of large scenarios with hundred or even thousands of virtual machines. This process uses the copy on write technique (cow), which starting the virtual machines from a single image file known as filesystem. In this way, the nodes can share the same filesystem. The filesystem is mounted in read-only mode. If a virtual node is modified, the differences are stored in a private filesystem.

VNX is also focused on education and research. In [15] a large virtual network scenario was created. It is a laboratory for dynamic routing that involve 44 virtual devices (16 Cisco routers, 6 Juniper routers, 6 Linux/Quagga routers, 12 end user and 4 Servers). This testbed is a typical scenario deployed with VNX and shows its potential.

One of the main SDN challenges is the integration of heterogeneous networks. VNX could provide the ideal environment to combine OpenFlow-enabled islands and legacy networks. The integration process is described in the next section.

IV. Integration and Validation

VNX should be implemented over a Linux operating system. The guidelines for configuration, modifications and filesystems are available in the official site of VNX project [27]. In order to testing with OpenFlow protocol, VNX needs the integration of some critical elements, an OpenFlow-enable switch for virtualization environments and a network operative system for network control. Additionally, it would be useful the integration of performance tools or data traffic analyzer. VNX was installed on a physical host with Ubuntu 12.04. Then, we choose two different filesystems. For controller device is desirable a graphical interface (ubuntu-12.04-gui-v024) to analyze the traffic. The second filesystem is a console interface (ubuntu-12.04-v024), which is used for simulated hosts and routers. The graphical filesystem was modified to make the controller functions, 3 of them were integrated: POX (based on Python) which is one of the most widely used today, NOX based on c++ and Python and finally Beacon which uses Java. The integration and configuration process are available in the official sites of each project. Additionally, in order to improve the functionalities of VNX/OpenFlow, three tools were installed: Wireshark, tcpdump and iperf. The wireshark tool is indispensable because originally it does not identify OpenFlow traffic. For this purpose, a dissector plugin for OpenFlow must be compiled and installed in the filesystem. Dissector allows to decode all information of specific incoming packets, in this case OpenFlow (version 1.0). Other important changes is the integration of Open vSwitch (OVS) [17]. OVS is an open source tool that allows the creation of switches in virtualization environments. OVS matches the virtual machines, providing better performance than the traditional bridge, such as VLANs, netFlow, QoS, bonding, mirroring, among others. OVS works transparently with VNX, for both legacy and OpenFlow networks. The version used in this paper is 1.4.0. After we create the .xml specification (Fig. 2).

```
<vm type="libvirt" name="C2" os="linux" subtype="kvm">
    <filesystem type="cow">/usr/share/vnx/filesystems/rootfs_ubuntu-gui</filesystem>
    <mem>512M</mem>
    <console display="yes" id="0"/>
    <console display="no" id="1"/>
  - <if id="2" net="Net1">
        <ipv4>10.0.1.3/24</ipv4>
    </if>
    <route type="ipv4" gw="10.0.1.1">default</route>
</vm>
<host>
  - <hostif net="Net2">
        <ipv4>10.0.2.2/24</ipv4>
    </hostif>
    <route type="ipv4" gw="10.0.2.1">10.0.0.0/16</route>
</host>
```

Fig. 2. XML Specification for Design Phase.

doi:10.15849/icit.2015.0037

Once we have the file with .xml specification, the virtual scenario is deployed and matched with the controller. For the validation process we replicate the topology of OpenFlow Tutorial, as a point of reference to see the VNX operation. This tutorial was developed by Stanford University [28] and it deploys a topology (subnet 10.0.0.0/24) with 3 virtual hosts (h2, h3 and h4), an OpenFlow switch (s1) and one controller (c0). Two scenarios are presented: a basic scenario (Fig. 3a) that is identical to OpenFlow Tutorial and the second scenario incorporates more subnets and a second controller (Fig. 3b).

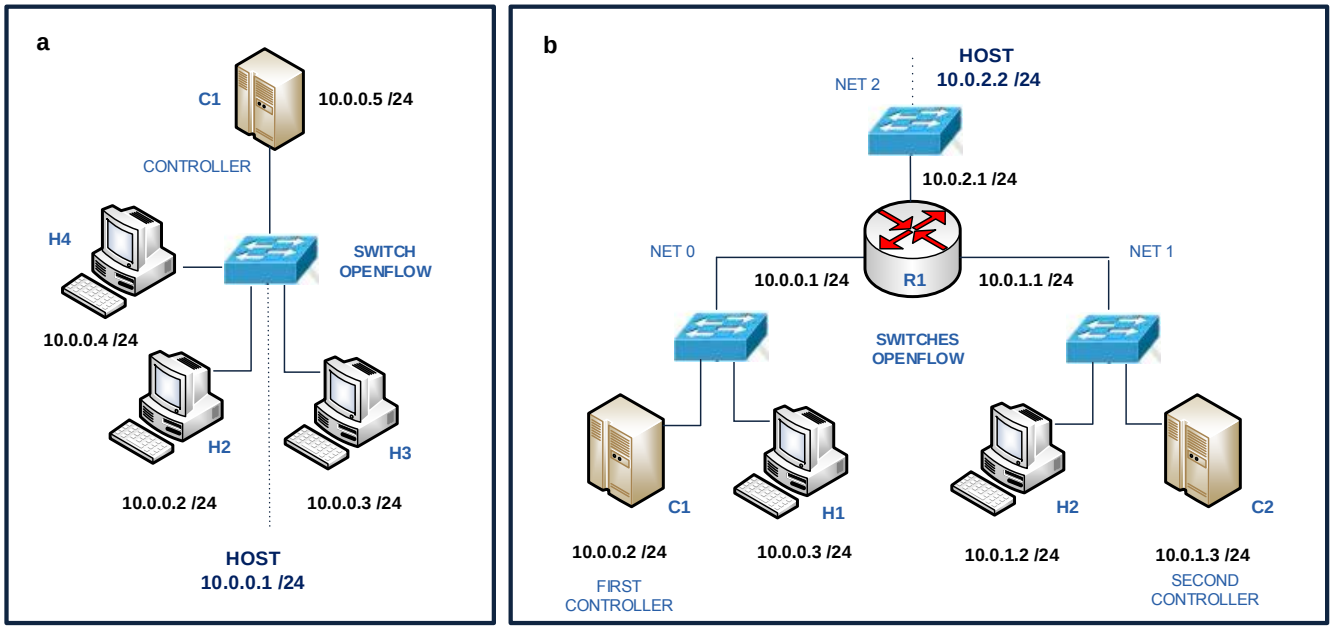

Fig. 3. (a) Scenario 1. Basic Scenario; (b) Scenario 2. Scenario with two Controllers.

The first scenario (Fig. 3a) has an OpenFlow-enabled switch and four hosts (C1, H2, H3, H4), all of them with Ubuntu 12.04. H2, H3 and H4 work with textual consoles and the controller (C1) works with a graphical console. The second scenario (Fig. 3b) is formed by five Ubuntu 12.04 virtual machines (router and hosts work with textual consoles and controllers with graphical console) according to the following structure:

- 3 switches in different subnets (Net0: 10.0.0.0/24, Net1: 10.0.1.0/24 y Net2: 10.0.2.0/24).
- 2 controllers (C1: 10.0.0.2 and C2: 10.0.1.3).
- 2 hosts (H1:10.0.0.3 and H2:10.0.1.2) each one in different subnets.
- Subnets communicate through the router (R1).

The proofs of concept of this work were made exclusively with Ubuntu virtual machines, but it is possible to use another kind of operating system. Data traffic was analyzed with Wireshark. At first, Wireshark shows only typical protocols, such as ICMP, UDP, IP, among others, because OVS works as an Ethernet switch by default.

In order to enable OpenFlow traffic, OVS must be connected with the controller. There are two configuration modes, which determine the switch behavior for a controller fail condition. These modes are:

- Fail standalone: The default configuration mode. If OVS does not receive the inactivity probe interval three times, the OVS takes the control of the switch and it works like a normal Ethernet switch (MAC-learning switch). When the connection is lost, the switch handles the incoming packets using the OFPP_NORMAL reserved port. Moreover, the switch will attempt to connect with the controller. These mode is usually available in OpenFlow hybrid switches.
- Fail secure: In this mode the OVS cannot take the network control if the controller fails. The network will be uncommunicated during the failure. Then, OVS will attempt to connect with the controller, until obtain a response. This mode is commonly used to avoid forwarding loops.

Once the communication is established, the controller (or controllers) should maintain the links with all switches. There are three kinds of roles for the connection. The default role is OFPCR_ROLE_EQUAL and it allows full control over the network. The second role is known as OFPCR_ROLE_SLAVE, in which switches are configured in read only mode, therefore the controller has limited control. The third role, OFPCR_ROLE_MASTER works in the same way that OFPCR_ROLE_EQUAL, but there is only one controller with this role, other controllers are changed to slave role. In the second scenario all switches are connected with C1 and C2 controllers in EQUAL role. In this way, we provide redundancy to the second scenario.

Proofs were made with standalone and secure mode in both scenarios. We used POX controller with three applications,

doi:10.15849/icit.2015.0037

forwarding.l2_learning, forwarding.l3_learning and forwarding.hub. Additionally, we wrote scripts in order to automate the process. These scripts contain the code for the deployment of the above mentioned scenarios and the establishment of links between switches and controller.

In both scenarios data traffic was generated with ICMP and web requests between the hosts of the topologies. OFP (message for the establishment of network communication), OFP-ARP, OFP-ICMP (packet-in, packet-out) messages were captured with Wireshark analyzer and tcpdump tools as shown in Fig. 4.

26	4.088734	10.0.0.2	10.0.2.2	OFP	74 Echo Reply
27	4.088827	10.0.0.2	10.0.2.2	OFP	74 Echo Reply
28	4.101774	02:fd:00:00:03:01	02:fd:00:00:01:01	OFP+ARP	126 Packet In
29	4.105569	10.0.0.2	10.0.2.2	OFP	90 Packet Out
32	4.106380	02:fd:00:00:01:01	02:fd:00:00:03:01	OFP+ARP	126 Packet In
33	4.108450	10.0.0.2	10.0.2.2	OFP	90 Packet Out
37	4.425297	10.0.0.3	10.0.1.2	OFP+ICMP	182 Packet In
39	5.038602	10.0.1.2	10.0.0.3	OFP+ICMP	182 Packet In
41	5.425545	10.0.0.3	10.0.1.2	OFP+ICMP	182 Packet In

Fig. 4. Traffic Capture Scenario 2.

Fig. 4 shows an ICMP proof from host h1 (10.0.0.3) to host h2 (10.0.1.2) performed in the second scenario, with the component forwarding.l2_learning of POX controller and in standalone mode.

Both scenarios work properly with OpenFlow protocol, however in second scenario there were duplicated messages (from controllers C1 and C2). This is because OpenFlow does not define coordination mechanisms among controllers in the same network or in different domains [29]. At present, this process is done with other components. For instance, Fonseca et al. in [30] introduces the CPRecovery component, which allows keeping the consistency between the primary and backup controllers. This component provides seamless transition between the primary and secondary controller through two steps, the replication phase (maintain updated data) and the recovery phase. The replication phase acts during the normal network behavior and the recovery phase acts in case of failure. Another challenge in large topologies is the communication among controllers in different SDN domains. At the present time, Internet Engineering Task Force (IETF) is working in a standard called interfacing SDN Domain Controllers (SDNi) for exchange routing information (network topology views, network conditions, event reports) and application requirement.

A general overview for the whole process in order to interact with VNX/OpenFlow scenarios is shown in figure 5. The first phase consist of the design and creation of VNX scenarios based on .xml specification. The second phase is related to the deployment or destruction of these scenarios through specific commands (vnx –f –v --create). Then, the controller must be connected with the switches and the user should configure the operation mode (standalone, secure, equal, slave, master). The user can create their own topologies and programs with the controller and finally can interact with the OpenFlow testbed.

V. Conclusion and Discussion

This work presents the integration process between VNX tool and OpenFlow protocol. The filesystems used by virtual machines and nodes was modified. We create a SDN environment through the integration of two main components: an OpenFlow compliant switch (Open vSwitch) and three network operating systems (NOX, POX and Beacon). Besides the controller has incorporated some performance and analyzer tools, these are Wireshark, tcpdump and iperf. Proofs of concept were carried out with POX components and two configuration modes (secure and standalone).

We can verified the exchange of OpenFlow messages (OFP+ARP, OFP+ICMP Packet In, OFP packet Out) with Wireshark analyzer. Although in the validation process we only used Ubuntu, future proofs can use multiple operating systems such as Windows. Now the user can create their own topologies and controller programs in order to experiment with OpenFlow protocol and SDN technology, which was the main objective of this work.

Today, VNX allows the deployment of large and complex OpenFlow networks in distributed environments. VNX allows not only the deployment of virtual scenarios in a single laptop, but also allows the inclusion of physical equipment (each one can have its own scenario with virtual machines), that is, VNX works in distributed scenarios. This is the main contribution of VNX over Mininet, since the communication between two scenarios in Mininet is a complex process. In this way, VNX enable the communication between OpenFlow networks and legacy networks that is one of the main challenges of SDN, the transition and migration process between heterogeneous networks. Besides, take into account that virtual scenarios may include Cisco and Juniper devices, therefore inside the virtual scenarios we could test with OpenFlow and no OpenFlow networks. Moreover, VNX allows the easy experimentation with specific services such as multimedia applications, deployment of servers, among others. The developer can customize the filesystem of the hosts and in this way, testing their new ideas and applications.

VNX also allows another kinds of operating systems for the virtual machines, such as Debian, Windows and Fedora. This is another strong point compared with Mininet, which uses only a Linux kernel. If a user want to test a Windows application over an OpenFlow network, the windows filesystem may include the application code.

doi:10.15849/icit.2015.0037

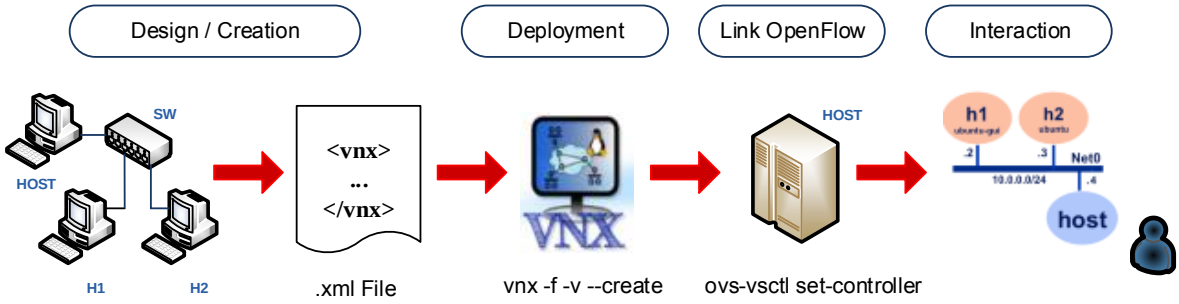

Fig. 5. Workflow of VNX/OpenFlow.

ACKNOWLEDGMENT

The research leading to these results has been partially funded by the European Union's H2020 Program under the project SELFNET (671672). Lorena Isabel Barona López and Ángel Leonardo Valdivieso Caraguay are supported by the Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación SENESCYT (Quito, Ecuador) under Convocatoria Abierta 2012 and 2013 Scholarship Program. This work was partially supported by the "Programa de Financiación de Grupos de Investigación UCM validados de la Universidad Complutense de Madrid – Banco Santander".

The authors would like to thank to David Fernández Cambronero for his comments and suggestions about VNX tool and Ana Lucila Sandoval Orozco for her valuable comments and suggestions to improve the quality of the paper.

REFERENCES

[1] W. Stallings, "Software Defined Networks and OpenFlow," *in The Internet Protocol Journal*, vol. 16, no. 1, March 2013, pp. 2-14.

[2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *in ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, April 2008, pp. 69-74.

[3] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, S. Shenker, "Ethane: Taking Control of the Enterprise," *in Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, New York, USA, August 2007, pp. 1-12.

[4] A. L. Valdivieso Caraguay, L. I. Barona López, L. J. García Villalba, "Evolution and Challenges of Software Defined Networking," *in Proceedings of the Workshop on Software Defined Networks for Future Networks and Services*, Trento, Italy, November 2013, pp. 47-55.

[5] O. S. Consortium et al., "OpenFlow Switch Specification v.1.3.4," March 2014 pp. 1-171.

[6] C. Elliott, "GENI: Opening Up New Classes of Experiments in Global Networking," *in IEEE Internet Computing*, vol. 1, January 2010, pp. 39-42.

[7] M. Suñé, L. Bergesio, H. Woesner, T. Rothe, A. Köpsel, D. Colle, B. Puype, D. Simeonidou, R. Nejabati, M. Channegowda, M. Kind, T. Dietz, A. Autenrieth, V. Kotronis, E. Salvadori, S. Salsano, M. Körner, S. Sharma, "Design and implementation of the OFELIA FP7 facility: The European OpenFlow testbed," *in Computer Networks*, vol. 61, March 2014, pp. 132-150.

[8] T. R. Henderson, M. Lacage, G. F. Riley, "Network Simulations with the ns-3 Simulator," *in Proceedings of the ACM SIGCOMM'08*, Seattle, WA, USA, August 2008, pp.17-22.

[9] POX, https://openflow.stanford.edu/display/ONL/POX+Wiki.

[10] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, S. Shenker, "NOX: Towards an Operating System for Networks," *in ACM SIGCOMM Computer Communication Review*, vol. 38, no. 3, July 2008, pp. 105-110.

[11] D. Erickson, "The Beacon Openflow Controller," *in Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, New York, NY, USA, August 2013, pp. 13-18.

[12] Floodlight project, http://www.projectfloodlight.org/.

[13] B. Lantz, B. Heller, N. McKeown, "A Network in a Laptop: Rapid Prototyping for. Software-Defined Networks," *in Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, New York, NY, USA, October 2010, pp. 1-6.

[14] S. Y. Wang, C. L. Chou, C. M Yang, "EstiNet OpenFlow Network Simulator and Emulator," *in IEEE Communications Magazine*, vol. 51, no. 9, September 2013, pp. 110-117.

[15] D. Fernández, A. Cordero, J. Somavilla, J. Rodriguez, A. Corchero, L. Tarrafeta, F. Galán, "Distributed Virtual Scenarios over multi- Host Linux Environments," *in Proceedings of the 5th IEEE International DMTF Academic Alliance Workshop on Systems and Virtualization Management*, Paris, France, October 2011, pp. 1-8.

[16] R. Sherwood, G. Gibb, K.-K. Yap, G. Appenzeller, M. Casado, N. McKeown, G. Parulkar, "Flowvisor: A Network Virtualization Layer," *in Technical Report OpenFlow Switch Consortium*, October 2009, pp. 1-15.

[17] B. Pfaff, J. Pettit, K. Amidon, M. Casado, T. Koponen, S. Shenker, "Extending Networking into the Virtualization Layer," *in Proceedings of the Eight ACM Workshop on Hot Topics in Networks*, HotNets-VIII, HOTNETS '09, New York City, NY, USA, October 2009, pp. 1-6.

[18] A. Kassler, L. Skorin-Kapov, O. Dobrijevic, M. Matijasevic, P. Dely, "Towards QoE-driven Multimedia Service Negotiation and Path Optimization with Software Defined Networking," *in Proceedings of the 20th IEEE International Conference on Software, Telecommunications and Computer Networks*, (SoftCOM), Split, Croatia, vol. 1, September 2012, pp. 1-5.

[19] R. Bennesby, P. Fonseca, E. Mota, A. Passito, "An Inter-AS Routing Component for Software-Defined Networks," *in Proceedings of the IEEE Network Operations and Management Symposium*, Maui, Hawaii, USA, April 2012, pp. 138-145.

[20] F. Farias, J. Salvatti, E. Cerqueira, A. Abelem, "A Proposal Management of the Legacy Network Environment Using Openflow Control Plane," *in Proceedings of the IEEE Network Operations and Management Symposium*, Maui, USA, April 2012, pp. 1143-1150.

[21] S. Das, G. Parulkar, N. McKeown, P. Singh, D. Getachew, L. Ong, "Packet and Circuit Network Convergence with OpenFlow," *in Proceedings of the IEEE Conference on Optical Fiber Communication (OFC), collocated National Fiber Optic Engineers Conference (OFC/NFOEC)*, San Diego, CA, USA, March 2010, pp. 1-3.

[22] M. Channegowda, R. Nejabati, M. Rashidi Fard, S. Peng, N. Amaya, G. Zervas, D. Simeonidou, R. Vilalta, R. Casellas, R. Martínez, "First Demonstration of an OpenFlow based Software-Defined Optical

doi:10.15849/icit.2015.0037

Network Employing Packet, Fixed and Flexible DWDM Grid Technologies on an International Multi-Domain Testbed," *in Proceedings of the European Conference and Exhibition on Optical Communication*, Amsterdam, Netherlands, September 2012, pp. 1-3.

[23] N. Handigol, S. Seetharaman, M. Flajslik, N. McKeown, R. Johari, "Plug-n-Serve: Load-Balancing Web Traffic using OpenFlow," *in Proceedings of ACM SIGCOMM Demo*, Barcelona, Spain, August 2009, pp. 1-2.

[24] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, W. Chou. "A Roadmap for Traffic Engineering in SDN-OpenFlow Networks". *in Computer Networks*, vol.71, June 2014, pp. 1-30.

[25] M. Kobayashi, S. Seetharaman, G. Parulkar, G. Appenzeller, J. Little, J. Reijendam, P. Weissmann, N. McKeown, "Maturing of OpenFlow and Software-defined Networking through Deployments," *in Computer Networks*, vol. 61, March 2014, pp. 151-175.

[26] F. Galán, D. Fernández, W. Fuertes, M. Gómez, J. E. L. de Vergara, "Scenario-based Virtual Network Infrastructure Management in Research and Educational Testbeds with VNUML," *in Annals of Telecommunications-Annales des Telecommunications*, vol. 64, May 2009, pp. 305- 323.

[27] "Virtual Networks over linuX (VNX)," http://web.dit.upm.es/vnxwiki/index.php/Main Page.

[28] OpenFlow Tutorial, http://www.openflow.org/wk/index.php/OpenFlow Tutorial.

[29] A. L. Valdivieso Caraguay, A. Benito Peral, L. I. Barona López, L. J. García Villalba, "SDN: Evolution and Opportunities in the Development IoT Applications," *in International Journal of Distributed Sensor Networks*, vol. 2014, May 2014 pp. 1-10.

[30] P. Fonseca, R. Bennesby, E. Mota, A. Passito, "A Replication Component for Resilient OpenFlow-based Networking," *in Proceedings of the IEEE Network Operations and Management Symposium, Maui*, Hawaii, USA, April 2012, pp. 933-939.

IET Networks

Review Article

IET Journals
The Institution of Engineering and Technology

Trends on virtualisation with software defined networking and network function virtualisation

ISSN 2047-4954
Received on 31st October 2014
Revised on 6th February 2015
Accepted on 2nd March 2015
doi: 10.1049/iet-net.2014.0117
www.ietdl.org

Lorena Isabel Barona López[1], Ángel Leonardo Valdivieso Caraguay[1], Luis Javier García Villalba[1] ✉, Diego López[2]

[1]*Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), Faculty of Computer Science, Office 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases 9, Ciudad Universitaria, 28040 Madrid, Spain*
[2]*Telefónica I+D, c/Don Ramón de la Cruz, 82-84, Madrid 28006, Spain*
✉ *E-mail: javierv@ucm.es*

Abstract: Software defined networking (SDN) and network function virtualisation (NFV) have become hot topics in recent years. On one hand, SDN decouples the control plane from the data plane allowing the rapid innovation and the introduction of new services in an easy way. On the other hand, currently proprietary appliances such as load balancers and firewalls are implemented in hardware, NFV aims to change these network functions to an open software environment using virtualisation and cloud technologies. This means a reduction of spends in the provisioning and management of telecom services. SDN and NFV are two different concepts but these can coexist and help each other. In this study, the authors present a survey of SDN and NFV focusing in virtualisation projects and the use cases where a synergy between these technologies is possible. This study includes the basic concepts of network virtualisation, NFV and SDN, current research and the relation between both technologies.

1 Introduction

Since Internet appeared in the 1970s, not only hardware equipment has been updated but also the appearance of new services. On one hand, the current services cover a broad range of topics such as cloud computing, real-time applications, video streaming, big data and social networks, among others. All of these services have triggered the exponential growth of network data traffic. However, the current architectures are not enough to cover these increased traffics in an efficient way and the standardisation process of new technologies takes a long time. On the other hand, there are many different proprietary devices, which are close and difficult to manage. When a service provider (SP) launches a different service it often needs the deployment of a new appliance, which increase both the operational (OpEx) and capital costs (CapEx). These issues have resulted in new concepts, such as software defined networking (SDN) and network function virtualisation (NFV), which are changing the way of managing and controlling networks.

SDN separates the data and the control plane in network devices in order to improve the programmability of the network [1, 2]. The SDN concept is not completely new, but rather, it takes the advances of three phases in network history [3]. First, active networks allow programmability capacities through the implementation of open interfaces in each network node. Second, the separation of the control and the data plane and finally the introduction of OpenFlow protocol [4] and network operative system (NOS) [5]. OpenFlow takes the common characteristic of traditional switches to create an open interface without exposing the internal structure of them.

Concerning hardware components, each time a new function is required, these are implemented in proprietary hardware. In this context, NFV is a novel approach based on virtualised network functions (VNFs), which aim to reduce the expenditures related to the deployment and management of new network applications. NFV increases the network elasticity because of that typical hardware appliances are changed for software functions that can run on homogeneous environment, breaking the constraints of proprietary hardware. These functions can be instantiated in different locations and can be deployed in the same standardised hardware [6].

SDN and NFV are two independent concepts but these are complementary between them. On one hand, SDN provides a full control of the network and automation capacities, which may eventually allow better performance of the virtual functions (VFs). Both technologies have common objectives such as faster time to market (TTM) for the deployment of new services and create independence from the hardware vendor. SDN/NFV could also enhance the current NFs providing better elasticity, scalability and programmability.

This paper presents a review of these technologies, their current development and the current challenges. This piece of work is organised as follows: Section 2 briefly gives an overview of network virtualisation history. In Section 3, we introduce the term NFV. Then, SDN approach is discussed in Section 4. In Section 5, we explain the relation between both technologies. Finally, in Section 6 we open a discussion and conclusion related with the topic.

2 Virtualisation network: a view in the history

Virtualisation is not a new concept; the first approach was the virtualisation of the computer's memory. Nowadays, it is possible to virtualise operative systems, computer hardware platforms, storage capacities and networks. Essentially, virtualisation concept refers to the abstraction of the logical resources from the physical resources, creating multiple logical instances over the same physical infrastructure [7]. In this sense, network virtualisation allows simultaneously multiple virtual networks (VNs) over a physical network as shown in Fig. 1.

There are some technologies that use the virtualisation principle within networks, such as virtual local area network (VLAN). VLAN is a logical network formed by a group of hosts in a single broadcast domain, where the logical topology is usually different from the physical network. VLANs introduce facilities for

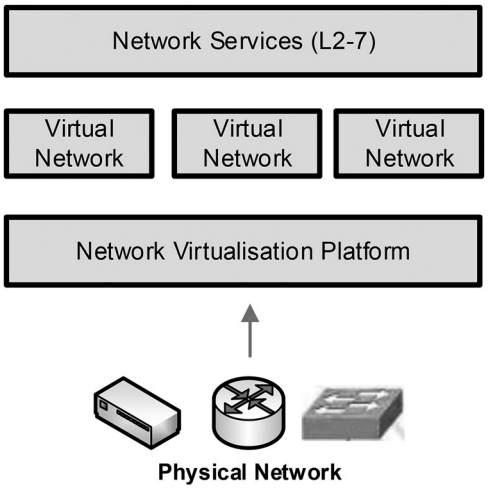

Fig. 1 *Network virtualisation*

management, reconfiguration and isolation of the network; however, the creation of 4094 VLANs is only possible over the same network. Virtual private network (VPN) is a VN that communicates multiple sites through a secured tunnel. These are geographically separated and the tunnel is carried over a public network [8]. There are different types of VPNs: layer 1 VPN (circuit switching domain), layer 2 VPN (L2 protocols such as Ethernet), layer 3 VPN (L3 protocols such as IP) and VPN over multi protocol label switching (MPLS). There are other approaches such as network interface card (NIC) virtualisation or Open vSwitch (OVS) [9]. OVS is an open source tool that allows the creation of switches in virtualisation environments. These switches allow the communication of virtual machines (VMs), providing better performance than the traditional bridge.

Other concepts such as active networks and overlay networks are also related to network virtualisation. Active networks control the network via an interface that exposes the nodes resources, allowing the automation and programmability of network devices. For its part, an overlay network allows the deployment of a logical network over the top of one or more physical infrastructures. The research community has generated many virtualisation projects in order to test with new technologies or applications such as global environment for network innovations (GENIs) or future Internet research for sustainable testbed (FIRST). GENI [8] provides a flexible platform in order to testing services and prototypes in computer networking and distributed systems. FIRST [10] is an experimental project in South Korea which promotes the research on future Internet architecture.

Recently, the virtualisation of the network components has given rise to novel concepts, such as the case of NFV. NFV applies virtualisation and cloud computing concepts in order to enhance the services provided by telecommunication SPs. NFV aims to design a standard networking device where any NF can be implemented, as is explained below.

3 NF virtualisation

NFV [6] is an initiative of main telecommunication SPs and European Telecommunications Standards Institute (ETSI), which made its first appearance in 2012. NFs industry specification group (NFV-ISG) includes more than 28 network operators and 150 enterprises from telecommunication industry. Nowadays, one of the main problems of telecom providers is the need for additional appliance when a new service is required. Often, these appliances are proprietary hardware, and therefore cannot be used by other providers. The investment in new hardware does not represent enough revenues if we take into account some factors such as the short life cycle of appliances, lack of space and complexity of the systems. NFV could help in these issues through the virtualisation of the NFs; these functions can run on standard switches, storage or high-volume servers. NFV may help in the following topics: accelerate the deployment of new services and NFs, faster TTM, reduce the energy consumption, capacity for multi-tenancy and multi-version, openness environment and reduce both CapEx and OpEx in the deployment and management of infrastructures and network services. At the same time, there are some challenges to overcome, these are: portability and mobility between vendors, high performance in virtualised appliances, security, elasticity, redundancy and compatibility with legacy appliances. NFV proposes the implementation of VNFs in software compatible with other platforms, allowing the rapid innovation and easy instantiation in several places. Fig. 2 describes the central idea of NFV.

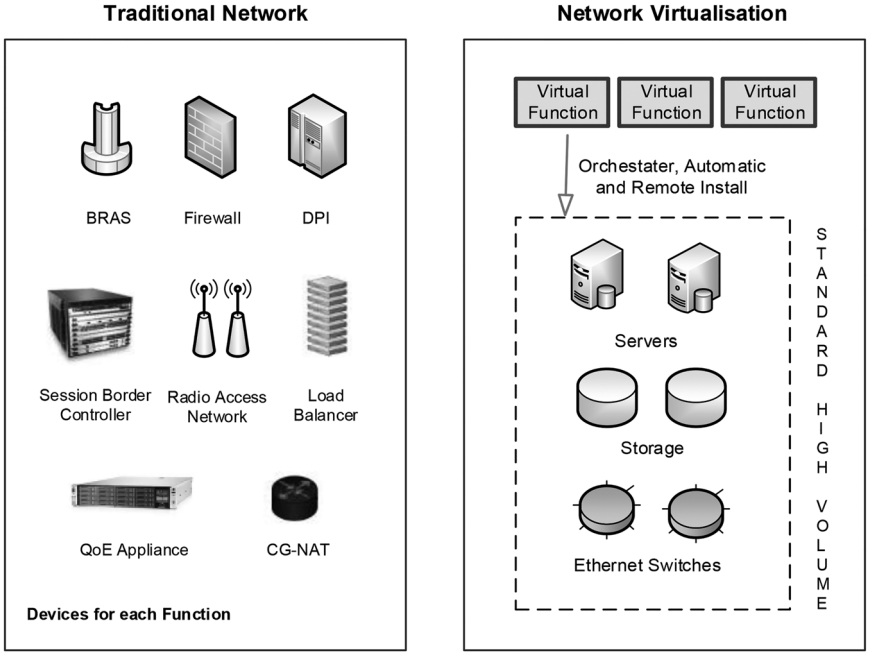

Fig. 2 *NFV approach*

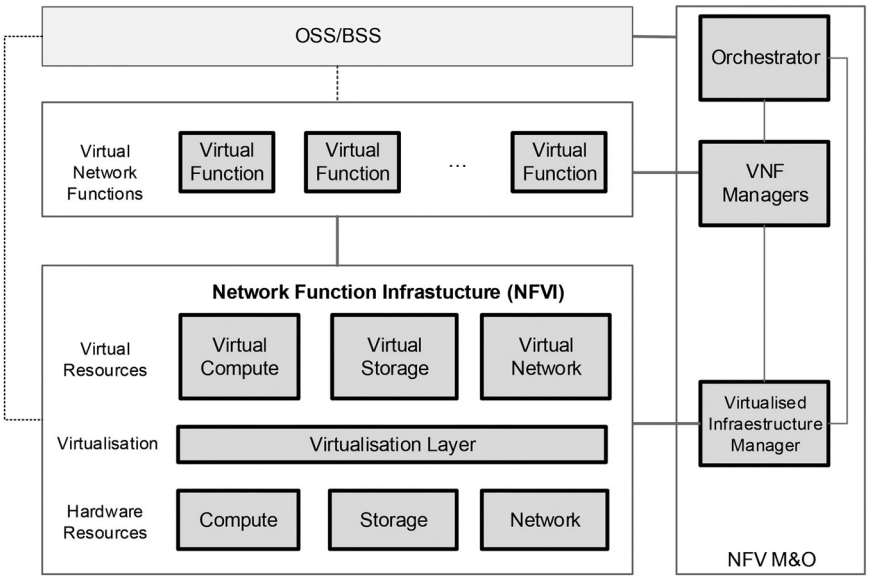

Fig. 3 *NFV architecture [11]*

NFV application fields mainly include switching elements, mobile network nodes, traffic analysis, service assurance, content distribution networks (CDNs), security functions and session border controller (SBC), among others. NFV uses the virtualisation concept of three hardware resources: computing, storage and network hardware. NFV architecture [11] is depicted in Fig. 3.

NFV identifies three main modules, these are:

- *NFV infrastructure (NFVI):* Represents the infrastructure of the system.

It includes the hardware resources (computing, storage and network), the virtualisation layer and the software instances of the hardware resources (virtual compute, virtual storage and VN). The virtualisation layer abstracts the hardware resources and ensures independency from different vendors and the deployment in different sites.

- *VNF:* Represents the implementation of an NF that runs over the NFVI. This module also contains the element management system, which manages the VNFs.
- *NFV M&O:* This component orchestrates and manages VNFs and NFVI. It has three elements, an orchestrator, a VNF manager and a virtualised infrastructure manager. Virtualised infrastructure manager has a resource inventory (computing, storage and network) in order to control the NFVI. Additionally, this module can also allow the integration with external business support system and operation support system (OSS) in order to bill the services rendered. Besides, this module guarantees the provisioning of services, no matter what the underlying technology.

There are a variety of NFV use cases defined by NFV-ISG [12]. Some of these cases are similar to services models of cloud computing (SaaS, infrastructure as a service and platform as a service). These cases are described below:

- *NFs virtualisation infrastructure as a service (NFVIaaS):* In this case, SPs can deploy its NFs in the infrastructure of another SP.
- VNF as a service (VNFaaS) allows the lease of VNFs. An enterprise can use a VNF of another SP. This case is similar to SaaS.
- In VN platform as a service (VNPaaS), an SP leases a set of applications and infrastructure (similar to platform as a service). The customer can introduce its own VNF.
- VNF forwarding graph (VNF-FG) creates a logical path with the hops (NFs) to deliver a service.

This case is useful to provide service chaining.

- Another use cases leverage the introduction of NFV in order to consolidate different types of network appliances to standard equipment, such as: virtualisation on mobile cloud network (virtualisation elements of evolved packet core), virtualisation of mobile base station (radio access network resources, eNodeB, worldwide interoperability for microwave access (WIMAX)), virtualisation of the home environment (virtualisation of customer premises equipment (CPE)) and the virtualisation of content delivery networks (CDNs).

These cases can be implemented with traditional mechanism, NFV, SDN or inclusive a combination of NFV with SDN. SDN can contribute to the network programmability and automation capacities in order to control VNs and functions, as is explained below.

4 Software defined networking

In recent years, we have seen a dramatic change in networks because of the exponential growth of network data traffic and the introduction of new technologies such as cloud computing or big data. To cover new requirements, protocols and standards must be created and tested before they are integrated in hardware. In conventional architectures, new protocols are included in a new software release or in dedicated hardware devices. Usually, this process takes a long time because all stakeholders must agree on the basic aspects related with the implementation and the deployment of new service. Once the standard is implemented, the network administrator must configure each device or change it; sometimes this process can take hours, days or even weeks, depending on the size of the network.

SDN takes into account the most outstanding contributions of the networking industry developed in the past 20 years. First, active networks introduce the programmability and automation of the network nodes (early 2000). Active networks allow the creation of customised services; however, this concept did not have a widespread use. There were performance problems and there was not a clear path for the deployment of these kinds of networks. Second, SDN applies the separating of control and data plane concept from projects such as ForCES (2001–2007) and finally it introduced two APIs (southbound and northbound) and NOSs (2007–present) [3].

In essence, SDN proposes a centralised control of the data plane, where the network intelligent resides in the controller (control plane). In this way, the manager can develop high-level applications to improve the network performance [4]. SDN architecture defines three main layers: application, control and data layers, as shown in Fig. 4.

SDN also defines two main APIs in order to connect these layers. Southbound API communicates hardware devices (data plane) with the controller (control plane) that, in turn, is connected to application layer through northbound API. Subsequently, SDN introduced East–West APIs for the communication between controllers in the same or a different domain.

Data layer is composed of the physical devices (switches and routers). The most well-known southbound interface is OpenFlow [4], that is, promoted by open networking foundation (ONF). ONF aims to accelerate the adoption of SDN and OpenFlow. It is comprised of over 100 members such as network operators, SPs and big companies such as Google.

OpenFlow switch (version 1.3.4) is based on the structure of traditional Ethernet switch. It takes the common characteristics from different vendors in order to programme the tables of the switch and handle packets based on a variety of packets header fields of different layers. The tables are divided in flow, group and meter tables [13]. The controller can add, delete or update flow entries in the flow tables. Each flow entry has associated match fields (for matching packets), counters (tracking packets), priority, timeouts and instructions (actions to be applied) in order to process the incoming packets. When a packet arrives, it is matched against flow entries of the first flow table (0) searching the highest priority. If a matching entry is found, the instructions are executed or may continue with the next tables, depend on the result of the match in the table. If the packet does not match with a flow entry in any table, the outcome depends on the configuration of the table miss flow entry (default rule).

For its part, the controller performs the rules that are applied on the switches. Nowadays, the NOS most widely used are: POX, Floodlight and the OpenDaylight (ODL) [2, 14] and all of them are open source projects.

ODL is an open source project promoted by Linux foundation which main objective is to foster the development and adoption of SDN applications. ODL introduces a plugin-based architecture that facilitates the centralised control and management of network in a flexible and modular way. ODL is implemented with Java and Python code and its structure is based on SDN architecture. ODL defines three layers: network applications and orchestration, the controller platform and the physical and VN devices. ODL has multiple southbound APIs for different protocols such as OpenFlow or Netconf and a bidirectional API based on representational state transfer (REST) (northbound API).

In some aspects, these NOSs are difficult to use, especially with the programming of complex functions. In this sense, northbound APIs such as Procera [15] and Frenetic [16] facilitate the creation of business applications or high-level programmes that are required by application layer.

There are several fields (wireless and mobile networks, security, quality of services (QoS), management, data centres and virtualisation) where SDN can be an enabler technology [17]. In particular, there are some successful migration cases from traditional network to an SDN such as Google Inter Datacenter-WAN and Stanford Campus Network, NTT Edge Case [18]. Indeed, network virtualisation has been a key to the success of these deployments, especially in data centres and cloud computing environments [19]. Currently, cloud computing infrastructure and large datacentres are composed of many racks, core and aggregation network devices and thousands of VMs. This paper is focused in the virtualisation concept and its applicability in SDN and NFV. Some SDN virtualisation projects are described below.

One of the first SDN approaches within network virtualisation is FlowVisor [20]. It allows the switch virtualisation to share the same network infrastructure with multiple tenants. It is based on the concept of computer virtualisation, that is, FlowVisor places a layer between data and control planes; in this way the network is divided in slices and each one can perform different tests without interference among them. FlowVisor uses the SDN concept to control each slice in order to test new research with production traffic. The slicing process takes into account four dimensions: the topology, the bandwidth, the device central processing unit (CPU) and the forwarding tables. To define the slice characteristics, the tenant uses a slice policy language to determine the network resources (fraction bandwidth and CPU), flowspace and the OpenFlow controller. Each slice has a text file composed of a list of tuples, and these in turn have specific actions (allow, read-only and deny).

FlowVisor rewrites the messages (from switch to controller and vice versa) to ensure the transparency between slices. Fig. 5 shows the FlowVisor architecture. FlowVisor is the base of many research and SDN deployments such as GENI and OFELIA [8, 18].

ADvanced FlowVisor (ADVisor) [21] is an enhanced version of FlowVisor that lets the creation of arbitrary topologies and it allows the sharing of the same flowspace between slices, which are the main limitations of FlowVisor. ADVisor is located between the hardware equipment and the controller and it directly sends the traffic to the respective slice, which are defined for the combinations of bits in the L2 field. For this purpose, ADVisor introduces two additional functions, virtual links management

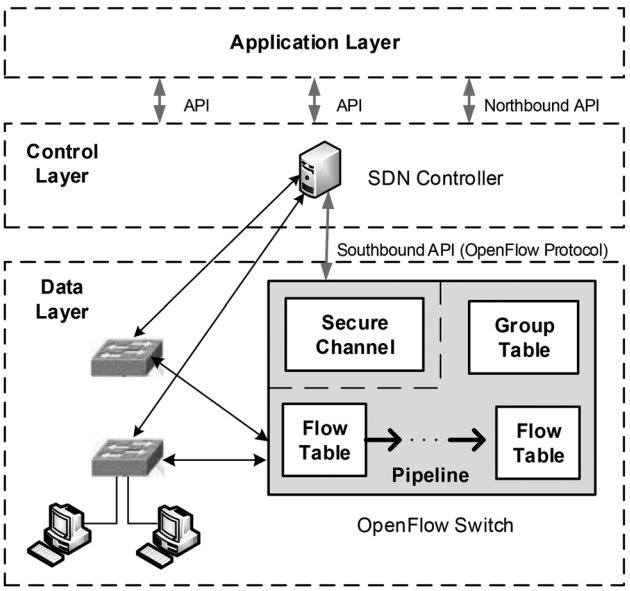

Fig. 4 *SDN architecture*

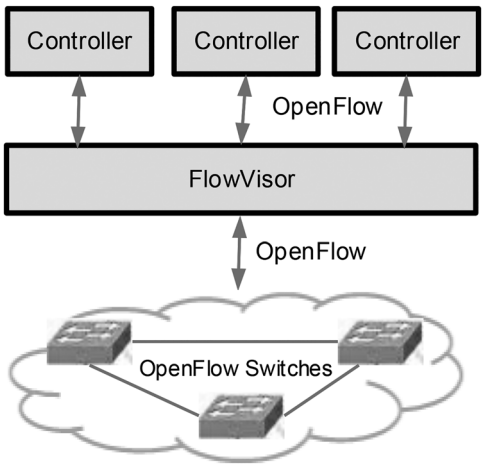

Fig. 5 *FlowVisor architecture*

(links between switches) and virtual ports management (connect virtual links with switches). In [22], a virtualisation framework is presented that allows simultaneous slices and different OpenFlow versions (OF 1.0–OF 1.1). Besides, it proposes enhanced functionalities related with QoS, management of the controller and the creation of SDN applications.

CloudNaaS [23] is a novel framework for cloud environments that allow the use of NFs such as isolation, QoS, custom addressing and transversal middleboxes. CloudNaaS lets fine-grained control over the network. It defines two main components, first the cloud controller which controls the virtual resources and the physical hosts. The second component is the network controller that manages the network devices. CloudNaaS uses four operations, first the customers specify their requirements through a simple policy language developed for this purpose. The second operation is to convert these policies in a communication matrix that is used to find out the best place for VMs. Then the matrix is converted to a network language which contains the rules to deploy the VMs. Finally, the rules are installed and configured in the network. Besides, CloudNaaS uses bin-packing for the placement of VM and allows the reuse of existing addresses.

Open application delivery networking platform (OpenADN) [24] is a design that combines the features of application SPs (ASPs) with the SDN benefits in order to allow the quick setup of application services in a distributed environment, looking such as a single data centre for each ASP. OpenADN is focused on cloud environment and uses current technologies such as OpenFlow, MPLS, session slicing and cross-layer communication. The project introduces the following elements: a virtualisation layer to slice the network, some NOSs, the network level control that invokes Internet service providers (ISPs) services and finally OpenADN creates and controls the applications for each ASP, as shown in Fig. 6. Besides, OpenADN introduces a label switching mechanism based on the composition of several sub-labels to allow the implementation of specific functions, for example, application label switching (Layer 3.5).

FlowN [25] is a system that allows network virtualisation providing an exclusive user space, arbitrary topologies and full control over the path for each tenant. FlowN is an extension of NOX controller and it uses advances in MySQL database to enhance the efficiency of the mapping process between the network and physical resources. FlowN uses an event handler to identify the packets that belong to each customer and it introduces the concept of container-based virtualisation to have multiple user-space containers independent of each other. The address space is defined by a set of fields of the packet headers and encapsulates incoming packets with an additional header (VLAN).

For its part, the relational database saves some aspects of the network: the physical topology, the virtual topology (nodes, interfaces and links) and the virtual–physical mapping. The virtual nodes can be a VM or an SDN-based switch and these are connected through some virtual interfaces and links. FlowN is able to establish some parameters such as the maximum number of flow table entries, the number of cores of the servers, bandwidth and latency for virtual links.

EHU OpenFlow enabled facility (EHU-OEF) [26] proposes a novel network virtualisation approach that enhances the flexibility and isolation between the slices. It is deployed in a real infrastructure, sharing resources between the production and the experimental traffic. EHU-OEF presents a modification of FlowVisor, enforcing slice isolation based on the MAC address settings, this method is known as Layer 2 prefix-based network virtualisation (L2PNV). EHU-OEF uses a modified NOX and allows a variety of headers such as VLANs. It also introduces an authentication and authorisation module.

VMware NSX [27] provides network virtualisation services (compute and storage) and security for data centres, where the user is able to deploy a VN as fast as a VM. NSX not only allows the deployment of L4–L7 network services but also the integration of additional capacities from third-party appliances such as specific load balancers, firewalls and so forth. NSX is based on the design of Nicira network virtualisation platform (NVP), which has an SDN-based architecture allowing the programmability of the VNs. NVP introduces a layer between the network and the final hosts. NSX components are depicted in Fig. 7.

Data plane contains an NSX vSwitch (vSphere distributed switch or OVS) which abstracts the physical network and allows communication with the hypervisor. Additionally, this plane uses a gateway between logical and physical networks (NSX Edge). The control plane has an SDN controller (NSX controller) and the management plane allows the configuration of vSphere environment. Finally, the consumption platform includes a cloud migration portal, which aids in the migration and the management in virtualisation and cloud environments. Another similar approach was developed by NEC, which is known as programmable flow (PFlow) [28]. PFlow is an SDN virtualisation solution that allows the deployment of multiple tenants in a secure environment. This paper was the first in introducing OpenFlow 1.3 and it supports OpenFlow 1.0.

A summary of current SDN virtualisation projects is presented in Table 1. It shows the applicability fields and the main characteristics of them.

As shown in Table 1, the target groups involve data centres, cloud environments, ISP, ASP, academy and research community. All of these projects ensure a degree of isolation between VNs and include OpenFlow 1.0. For its part, flexibility is related to some factors such as scalability, dependence of specific technologies and deployment facilities. High scalability requires more resources in order to support more tenants or virtual applications; this is limited by the memory size, the number of forwarding flows and bandwidth, among others. Moreover, projects carried out by enterprises [27, 28] provide modules to guarantee QoS, CloudNaaS [23] guarantees bandwidth for each VN and Advisor [21] guarantees address space.

The network virtualisation approaches are based on three methods: first, improving the functionalities of the controller [23, 25], second using an enhanced version of FlowVisor [21, 22, 26]

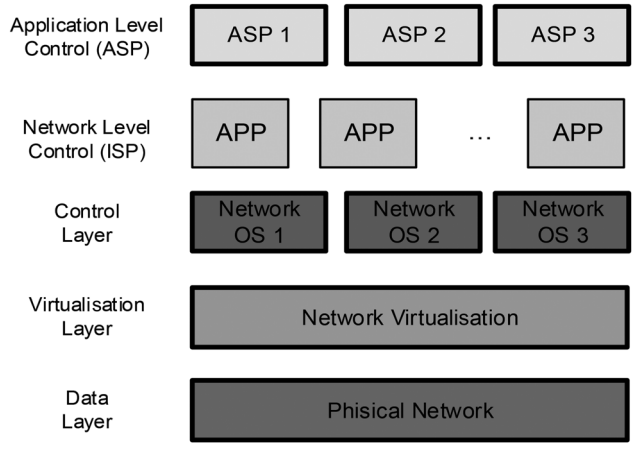

Fig. 6 *OpenADN architecture [24]*

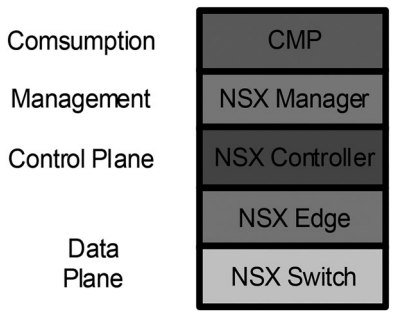

Fig. 7 *VMware NSX components*

Table 1 Characteristics of SDN virtualisation projects

Project	References	Domain	Features	Resource virtualised	ISO[a]	Flex[b]	QoS
FlowVisor	[20]	network testbed and simulation, academy research	line rate deployment, OVS	bandwidth, topology, traffic device CPU, forwarding tables	yes	not	not
ADVisor	[21]	enhanced FlowVisor, academy research	MPLS, NOX controller	traffic, address space, topology, bandwidth	yes	not	yes
CloudNaaS	[23]	network as a service, cloud providers	OVS, NOX controller	compute resources, storage resources	yes	not	yes
OpenADN	[24]	data centres, ASPs cloud providers, ISPs	MPLS, label switching 4–7 middleboxes, cross-layer communication	compute resources	yes	yes	not
FlowN	[25]	data centres, cloud providers	container-based virtualisation based on NOX controller	topology, bandwidth device, CPU (cores)	yes	yes	not
EHU-OEF	[26]	real testbeds, academy and research	L2PNV (layer 2 prefix-based network virtualisation), NOX controller	traffic, forwarding tables memory, interfaces	yes	yes	not
VMware NSX	[27]	datacentres, overlay networks ISPs public and private enterprises	OVS, vSphere distributed switch OVS, neutron	compute resources, storage resources	yes	yes	yes

a Isolation guarantee.
b Flexibility (deployment facilities and scalability).

and third creating a new component [26–28]. SDN can be benefited by the contribution of new concepts; it is the case of NFV.

5 Relation between SDN and NFV

As previously mentioned, NFV and SDN aim to break the innovation barrier between proprietary appliances and thus accelerate the introduction of new services. It is important to note that SDN is focused on maximising the network resources and NFV the storage and server resources. Fig. 8 shows the relation between SDN and NFV presented by ONF [29].

SDN maintains its structure based on three layers with OpenFlow protocol as the southbound API. The controller proposed is ODL, which supports not only OpenFlow but also another different southbound APIs. This overview presents open northbound APIs based on OpenStack [30] and NFV. Both technologies could control the compute, storage and networking resources in an efficient way. Finally, the application layer proposes the use of NFs in order to create network applications or new services.

The Network Virtualisation Report of 2014 [31] presents the main trends in this field and the SDN/NFV ecosystems created by the main vendors, such as Avaya, Big Switch Networks, NEC, Cisco Systems and ConteXtream, among others. For instance, ConteXtream Corporation realises SDN/NFV demonstrations related with virtual evolved packet core (EPC) and subscriber –aware service function chaining, load balancing and monitoring functionalities. In the same way, Huawei and Cisco have shared the latest progress in SDN/NFV deployments.

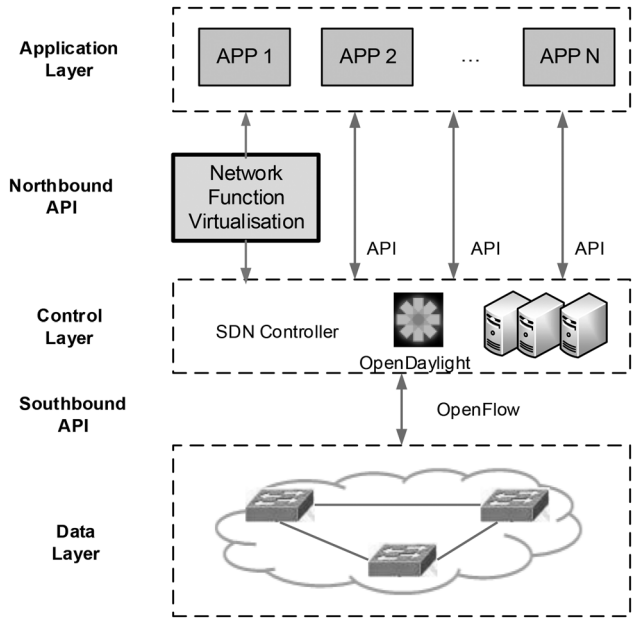

Fig. 8 *SDN/NFV*

In such a manner, at the end of the year 2014, ETSI announced the launch the open platform for NFV project (OPNFV) [32], which is based on referential architecture defined by ETSI NFV-ISG [6]. OPNFV will aid to accelerate and promote NFV concept in a new platform based on open source tools (KVM and OVS), cloud technologies (OpenStack and CloudStack) and SDN projects (ODL controller). The research community may develop, deploy and test their own NFs in order to enhance the availability, scalability and reliability of services provided by telecom providers. This initiative is the first open source project within NFV and its first software release is planned for 2015.

The applicability of SDN has gained the attention of industry and research community not only in wired networks but also in mobile networks (software defined wireless network and cellular networks) [2]. In [33] are illustrated the challenges and the potential advantages of SDN in this area such as the virtualisation or the mobility control. In the wireless field, there are real testbeds such as OpenRoad [34] (wireless fidelity (Wifi) and WIMAX nodes). Other approaches allow the deployment of radio base stations in the cloud [35] or the improvement of the core of cellular networks such as the CellSDN project [36].

The challenges and requirements to integrate NFV in mobile networks are presented in [37]. In particular, the case of virtualisation of the EPC (vEPC). It analyses and divides the vEPC components in order to achieve better control and less congestion in data plane. This approach takes into account four segments, the first mobility management entity (MME) with home subscriber server (HSS) (enhance authentication and authorisation process), the second service GPRS support node (SGSN) with home location register (HLR) (support combined systems), the third segment consists of packet data network gateway (PGW), policy and charging enforcement function (PCEF) and serving gateway (SGW) (centralised processing in the data plane) and finally user data respository (UDR), policy and charging rules function (PCRF), online charging system (OCS) and off line charging system (OFCS) (Unify user database, less fragmentation).

The industry and research community go a step further in this direction and propose the combination of SDN and NFV [38]. This combination might provide a standardised environment, where the introduction of new services might be done in less time, with the lowest investment and allowing the easy integration of old and new techniques and technologies. However, there are

some concerns related to the widespread use of SDN/NFV in mobile networks. First, there are not consensus in the way that SDN concept is applied in these kinds of networks, if is compared with the OpenFlow wire deployments. The other important concern is that NFV is still in early stage. Although both technologies will not fully integrate in the short term, it is clear that an architecture based on both technologies could be a referential point to telecom operators, as evidenced by ETSI NFV report [6]. This report presents some use cases that combine SDN and NFV [29]. First, VNF-FG applies the SDN concept to control the chaining process of virtual appliances in a dynamic way (add, delete and update), reducing the deployment times from weeks to minutes. Chaining also organises multiple VNFs in sequence in order to deliver a service.

In NFIaaS case, an SP 'A' can offer a specific service to the customers in sites where it does not have geographic coverage but the SP 'B' does. With a centralised control, the monetisation and the management process is more efficient and both SPs obtain revenues, the first with the service and the second with the infrastructure lease. In the case of the virtualisation of the CPE, the SP has the remote control of the devices and SDN can let a reliable connection. Traditional CPE may be replaced by a VF; in consequence the OpEx is reduced. It will not be necessary that an operator goes to the physical site in where CPE is placed. For its part, some functions such as deep packet inspection (DPI) may be benefited by NFV/SDN. Nowadays, DPI is used in a variety of applications such as gateways, load balancers, policy control and so on. A DPI virtualised function may provide a standard appliance in order to supervise the network and it reduces the capital investment.

Some projects combine both technologies, for instance, EmPOWER [39] shows a testbed composed of 30 nodes in the University of Trento. This facilitates the deployment of SDN/NFV experiments for Wifi networks and it also provides monitoring tools in order to control the energy consumption. EmPOWER can support high-level programming primitives, a set of primitives related with the network status and an interface for the service instantiation such as a web-based control framework or a command line interface. EmPOWER architecture consists of a single master agent (implemented within Floodlight controller) and some agents for access points (APs). The network services run on top of the controller and can use a Floodlight REST interface or an interpreter such as Pyretic. The main objective of EmPOWER is to test energy aware mobility management schemes in real infrastructure. It is also able to shut on/down the APs, depends on the capacity utilisation.

In the same way, Batallé *et al.* [40] designs a routing NF based on OpenFlow protocol and NFV. This paper shows some use cases: migration of IPv4 to IPv6 and inter domain routing. This NF is formed by three components. First, the controller module that analyses and controls the data traffic. The second module contains the routing function that receives instructions from controller. Finally, the third component is in charge of the communication of aforementioned modules. The proposal is based on a modified version of Floodlight controller (NFV module) and the development of the routing function as an OpenNaaS resource.

For its part, Italtel R&D in [41] presents the advances in NFV/SDN specifically the SBC. SBC was fully virtualised in order to control the interconnection between two networks. Virtualised SBC was deployed over the carrier grade Linux operating system of Italtel and it includes proprietary software application. The hypervisor is based on VMware vSphere Hypervisor 5.1. The testbed includes two Cisco servers, in which three VMs run, and each of one with a three different functions.

Cannistra *et al.* [42] present an SDN/NFV testbed that consists of three data centres in a ring topology. It uses OpenFlow protocol, Floodlight controller and distributed overlay virtual Ethernet (DOVE). Some applications were created such as a graphical user interface (Avior), which controls the OpenFlow appliances (monitor statistics and firewall) with mobile devices such as smart phones or tablets. The testbed probes a VM migration with a 75% of server utilisation. The applications continue without interruptions during the migration; however, there were problems with dynamic workloads. The VM migration takes into account some parameters: VM memory size, page dirty rate per second and network bandwidth.

Network operative system (NetOS) [43] combines SDN with NFV in order to obtain a mentality based on software, where any VFs can be added, managed, moved and updated efficiently and in an easy way, no matter what the kind of commodity hardware, the vendor, their place and other variables. It supports OpenFlow protocol and it can interact with the components of NFV architecture. NetOS defines three main components: (i) 'drivers and devices' which contains a network abstraction layer which provides an unified southbound API to the physical appliances. (ii) 'NetOS kernel' which maintains the whole network state and it has a virtualisation network layer and (iii) 'user space' which facilitates the integrations with OSS and controller.

In [44], a reference model that combines SDN with NFV is presented. It proposes a network abstraction model (NAM) that allows for the creation of a single framework. NAM takes into account three requirements: first, it should allow new or existing NFs. Second the framework should be extensible, which requires the accommodation of new SDN/NFV blocks and third NAM should be expressive between the interfaces and the controller. NAM also defines two types of interfaces: configuration and management interfaces. In the same way, Masutani *et al.* [45] present the characteristics that a network node should cover in order to support NFV technology. This project provides the initial prototype of a virtual broadband remote access router (BRAS); implemented with Intel DPDK, OVS and OpenFlow protocol. The use cases cover two kinds of services: Internet connection services and SIP services.

OpenSDNCore [46] is an initiative developed by Fraunhofer FOKUS that creates a prototype implementation to experiment with the capabilities of SDN and NFV technologies. It is aligned with OpenStack and OpenFlow 1.4 in order to support telecom features. Additionally, this prototype integrates with other projects such as OpenEPC and OpenIMSCore.

Furthermore, UNIFY [47] provides an environment to deploy NFs based on the combination of cloud computing and virtualisation. The project intends to design a universal hardware node to deploy these functions in an open environment while decrease the deployment and management costs. UNIFY supports a variety of technologies such as OpenFlow and NFV and is focused on three areas: infrastructure virtualisation, flexible service chaining and finally network service chain invocation. Moreover, T-NOVA project [48] aims to design and implement an NFV/SDN framework to deploy VNFs. This NFs are developed in software and eliminate the need of acquire, install and maintain specialised hardware. The framework will create a marketplace wherein the developers could offer their NFs while allowing customers use these applications or services.

Table 2 shows the initial deployments and tests in order to probe the benefits that telecom industry could obtain from the combination of SDN and NFV.

Current projects try to tackle current needs in telecom environments, especially in fifth generation (5G) networks. An example of which is Mobile and wireless communications Enablers for the Twenty–twenty Information Society (METIS) [49] or Connectivity management for eneRgy Optimised Wireless Dense (CROWD) projects [50]. CROWD networks propose an architecture to enhance the performance on very dense and heterogeneous wireless networks (Dense Nets). CROWD provides dynamic controller placement, dynamic backhaul reconfiguration, energy and MAC optimisation and ensures user quality of experience. For this purpose, this project uses OpenFlow to control and manage the nodes (eNBs and Wifi AP).

METIS [51] aims to break the barriers within 5G mobile and wireless communication system and lay the bases for the standardisation process. It is focused on current and future needs of mobile networks such as low latency, high data rate, low energy consumption, ubiquitous communication, massive machine communication, ultra dense networks and better QoS, among

Table 2 SDN/NFV projects

Project	References	Domain	Description	Features
ConteXtream	[31]	SDN/NFV VN environment	it presents a solution that implements network virtualisation for carrier providers	vSphere environment, multi-hypervisor support, any cloud platform Virtual eXtensible Local Area Network (VxLAN), stateless transport tunneling (STT), Generic Routing Encapsulation (GRE) ODL
EmPOWER	[39]	SDN/NFV testbed	testbed within wireless domain, composed by 30 nodes (Trento). Use cases include resource utilisation and dynamic handover Wifi	FlowVisor, Floodlight Pyretic, OVS, light virtual AP Energino, Click Modular Router
Batallé *et al.*	[40]	routing function virtualisation	it designs a virtual routing function. This function is tested in Mininet simulator	Floodlight, OpenNaaS, Mininet, OVS
Monteleone and Paglierani	[41]	SBC based on SDN/NFV	it proposes a virtualised NF of SBC deployed in Italtel R&D laboratory	carrier grade Linux, operative system of Italtel, VMware vSphere, Hypervisor Cisco Servers
Cannistra *et al.*	[42]	SDN/NFV testbed	three data centres connected in a ring topology (125 km). Use cases include probes to VM migration	OpenFlow, Floodlight, DOVE
NetOS	[43]	SDN/NFV platform	it presents an initial approach in order to provide an SDN/NFV environment (NOS)	OpenFlow, simple network management protocol (SNMP) not specify more tools
Haleplidis *et al.*	[44]	SDN/NFV reference model	it proposes a single framework that combines SDN and NFV	it takes into account: ForCES and Click Modular Router
Masutani *et al.*	[45]	virtual BRAS	it presents an initial design of a virtual BRAS using Intel DPDK	physical or cloud networks, OVS, Linux new application programming interface (NAPI) model. kernel-based virtual machine (KVM) hypervisor
OpenSDN Core	[46]	prototype of SDN/NFV	environment that allows the experimentation with SDN and NFV. Some used cases could include vEPC, vIMS and vRAN	OpenFlow 1.4 OpenStack GTP GRE
Unify	[47]	architecture to NFs	it proposes an architecture to flexible creation and deployment of NFs	OpenFlow, Intel, data plane development kit (DPDK) OpenStack
T-Nova	[48]	NFaaS over virtualised infrastructure	it provides a framework that allows the easy deployment of NFs. It provides an NF marketplace	OpenFlow, OpenStack

others. To cover these needs, METIS might progress on the following areas: radio-links, multi-node/multi-antenna technologies, multi-layer and multi-RAT networks and spectrum usage. METIS also creates new opportunities in five scenarios, for which establish a set of requirements and key performance indicators. These scenarios are:

- 'Amazingly fast' (provide very high data-rates without connectivity delays).
- 'Great service in a crowd' (provide QoE in crowd places such a massive events).
- 'Best experience follows you' (provide services to users on movement such as cars).
- 'Super real-time and reliable connections' (provide reliability and low latency in new applications, such as M2M).
- 'Ubiquitous things communicating' (give services to machines devices).

The advances obtained from METIS could be enhanced with the integration of an SDN/NFV approach. One of the main concerns is the capacity to deploy new services no matter the SP or the location, characteristic that might be covered by NFV. Moreover, SDN has showed the improvement of the network management.

Even though METIS project could be a referential point in 5G technologies, SDN/NFV not only addresses this field but also other areas such as cloud computing and data centres, among others [52]. Indeed, advances in 5G networks have become a prioritised topic in the research agenda, as evidenced the European Commission with the call 'Advanced 5G Network Infrastructure for the Future Internet' [53]. This aims to encourage competitiveness of the network operators within 4G/5G networks by means of novel technologies such as SDN and NFV.

6 Discussion and conclusion

Today, the number of services is growing faster than ever before, and customers want to use these services almost immediately. SDN/NFV could shorten the lifecycle in the development and innovation of new applications. On one hand, the research community aims to accelerate the testing process in order to introduce a new technology. On the other hand, SPs and networking enterprises aim to accelerate the TTM new services and reduce the expenditures, in consequence the final customer could obtain enhanced services.

The lifecycle of network devices includes two important phases: first, the deployment and allocation of resources and second the control and management of the devices. SDN allows a better control of the network, giving programmability capacities and improved network management. SDN also encourages the innovation and facilitates the automation of the network. For its part, NFV increases the network flexibility and reduces the complexity in the deployment of traditional NFs. The new VFs could be provided as a software component.

NFV and SDN are independent and complementary at the same time but the synergy of both technologies could provide an open environment to foster the innovation and decrease the capital and OpEx related to new infrastructure and services, all of this with major control and automation of network resources. For instance, the combination of SDN with NFV enables the dynamic service chaining through its centralised controller. The packet processing can be done as an NF and the controller can programme the flow table in the switch based on subscriber awareness of a given flow. Other application case is related to the deployment of VNs, NFV is responsible of the creation and SDN of the management of them.

There are some challenges to overcome due to the fact that SDN and NFV are involved in a maturing process. The main challenges are related to the migration of VFs between different sites and vendors (mobility and portability), NF management, service continuity and authentication and authorisation methods because of the security problems that introduce new layers. One of the most important challenges is the need to upgrade the current software of the network devices or in the worst case, the change of the entire infrastructure. All stakeholders should create solutions that allow the easy expansion and management of the network devices and applications. Furthermore, it is important to design new reliability and redundancy schemes in order to guarantee the network performance. First, methods to schedule the resources, second the creation of redundancy mechanisms to recover the network status after a failure and third mechanisms for the coexistence with legacy networks. It is important to note that

mobile and sensor networks need a better control and manage of data traffic. CROWD and METIS provide some ideas to enhance these kinds of networks; however, there are some problems related to the mobility and resource constraints.

The networking industry is changing towards a software-driven model deployed on standard hardware. This model could align with the current services and products, allowing the customisation of the services and fast scalability, hence accelerating the innovation and reduce the expenditures. This paper presents a chronology of network virtualisation and the description of two key technologies, SDN and NFV. It presents the current projects and applications of SDN in virtualisation field. It also describes the NFV concept and its use cases, as well as the relation between these technologies. This paper is intended to explain the benefits that we could obtain from the combination between SDN and NFV and the challenges embracing this topic.

7 Acknowledgment

This work has been partially supported by the European Commission Horizon 2020 Programme under grant agreement number H2020-ICT-2014-2/671672 - SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks). Lorena Isabel Barona López and Ángel Leonardo Valdivieso Caraguay are supported by the Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT), Quito - Ecuador.

8 References

1 Lara, A., Kolasani, A., Ramamurthy, B.: 'Network innovation using OpenFlow: a survey', *IEEE Commun. Surv. Tutor.*, 2013, **16**, (1), pp. 493–512
2 Valdivieso Caraguay, Á.L., Benito Peral, A., Barona López, L.I., García Villalba, L.J.: 'SDN: evolution and opportunities in the development IoT applications', *Int. J. Distrib. Sens. Netw.*, 2014, **2014**, pp. 1–10
3 Feamster, N., Rexford, J., Zegura, E.: 'The road to SDN: an intellectual history of programmable networks', *ACM SIGCOMM Comput. Commun. Rev.*, 2014, **44**, (2), pp. 87–98
4 McKeown, N., Anderson, T., Balakrishnan, H., *et al.*: 'OpenFlow: enabling innovation in campus networks', *ACM SIGCOMM Comput. Commun. Rev.*, 2008, **38**, pp. 69–74
5 Valdivieso Caraguay, Á.L., Benito Peral, A., Barona López, L.I., García Villalba, L.J.: 'Evolution and challenges of software defined networking'. Workshop on Software Defined Networks for Future Networks and Services, 2013, pp. 47–55
6 ETSI Industry Specification Group (ISG): 'Network function virtualization (NFV) white paper'. SDN and OpenFlow World Congress, Frankfurt-Germany, 2013, pp. 1–16
7 Jain, R., Paul, S.: 'Network virtualization and software defined networking for cloud computing: a survey', *IEEE Commun. Mag.*, 2013, **51**, (11), pp. 24–31
8 Chowdhury, N.M., Boutaba, R.: 'A survey of network virtualization', *Comput. Netw.*, 2010, **54**, (5), pp. 862–876
9 Pfaff, B., Pettit, J., Amidon, K., Casado, M.: 'Extending networking into the virtualization layer'. Proc. of ACM SIGCOMM HotNets, ACM, 2009, pp. 1–6
10 Luo, M.Y., Chen, J., Mambretti, J., *et al.*: 'Network virtualization implementation over global research production networks', *J. Internet Technol.*, 2010, **14**, (7), pp. 1061–1072
11 ETSI Industry Specification Group (ISG): 'Network function virtualization (NFV) architectural framework', October 2013, pp. 1–21
12 ETSI Industry Specification Group (ISG): 'Network function virtualization (NFV) use cases', October 2013, pp. 1–50
13 Consortium, O.S.: 'OpenFlow Switch Specification v.1.3.4', 2014, pp. 1–171
14 'OpenDaylight: Technical Overview'. Available at http://www.opendaylight.org/project/technical-overview, accessed February 2015
15 Voellmy, A., Kim, H., Feamster, N.: 'Procera: a language for high-level reactive network control'. Proc. of the First Workshop on Hot Topics in Software Defined Networks, 2012, pp. 43–48
16 Foster, N., Guha, A., Reitblatt, M., *et al.*: 'Languages for software defined networks', *IEEE Commun. Mag.*, 2013, **51**, (2), pp. 128–134
17 Lee, S.W., Han, S.W., Kim, J.W., *et al.*: 'FIRST: Korean future Internet testbed for media-oriented service overlay network architecture', *J. Internet Technol.*, 2010, **11**, (4), pp. 553–559
18 Kobayashi, M., Seetharaman, S., Parulkar, G., *et al.*: 'Maturing of OpenFlow and software-defined networking through deployments', *Comput. Netw.*, 2014, **61**, pp. 151–175
19 Courtney, M.: 'Service as a standard [software-defined networking concept]', *IET Eng. Technol.*, 2012, **7**, (11), pp. 64–67
20 Sherwood, R., Gibb, G., Yap, K.K., *et al.*: 'Can the production network be the testbed?'. OSDI, 2010, vol. 10, pp. 1–6
21 Salvadori, E., Corin, R.D., Broglio, A., Gerola, M.: 'Generalizing virtual network topologies in OpenFlow-based networks'. Global Telecommunications Conf., 2011, pp. 1–6
22 Sonkoly, B., Gulyás, A., Németh, F., *et al.*: 'OpenFlow virtualization framework with advanced capabilities'. Software Defined Networking (EWSDN), 2012, pp. 18–23
23 Benson, T., Akella, A., Shaikh, A., Sahu, S.: 'CloudNaaS: a cloud networking platform for enterprise applications'. Proc. of the Second ACM Symp. on Cloud Computing, NY, USA, 2011, pp. 1–13
24 Paul, S., Jain, R.: 'OpenADN: mobile apps on global clouds using OpenFlow and software defined networking'. Globecom Workshops, 2012, pp. 719–723
25 Drutskoy, D., Keller, E., Rexford, J.: 'Scalable network virtualization in software-defined networks', *IEEE Internet Comput.*, 2013, **17**, (2), pp. 20–27
26 Matias, J., Mendiola, A., Toledo, N., Tornero, B., Jacob, E.: 'The EHU-OEF: an OpenFlow-based layer-2 experimental facility', *Comput. Netw.*, 2014, **63**, pp. 101–127
27 'VMware® NSX Network Virtualization Design Guide'. Available at https://www.vmware.com/products/nsx/, accessed February 2015
28 'Programmable Flow'. Available at https://www.necam.com/sdn, accessed February 2015
29 Open Networking Foundation (ONF): 'OpenFlow-enable SDN and network function virtualization', February 2014, pp. 1–12
30 'OpenStack: The Open Source Cloud Operating System'. Available at http://www.openstack.org, accessed February 2015
31 'Network Virtualization Report 2014 Edition', SDNCentral. Available at http://www.sdncentral.com, accessed February 2015
32 'Open Platform for NFV Project (OPNFV)'. Available at https://www.opnfv.org/, accessed February 2015
33 Bernardos, C.J., De La Oliva, A., Serrano, P., *et al.*: 'An architecture for software defined wireless networking', *IEEE Wirel. Commun.*, 2014, **21**, (3), pp. 52–61
34 Yap, K.K., Kobayashi, M., Sherwood, R., *et al.*: 'Empowering research in mobile networks', *ACM SIGCOMM Comput. Commun. Rev.*, 2010, **40**, (1), pp. 125–126
35 Haberland, B., Derakhshan, F., Grob-Lipski, H., *et al.*: 'Radio base stations in the cloud', *Bell Labs Tech. J.*, 2013, **18**, (1), pp. 129–152
36 Li, L.E., Mao, Z.M., Rexford, J.: 'CellSDN: software-defined cellular networks'. Technical Report, Princeton University, 2012, pp. 1–6
37 Hawilo, H., Shami, A., Mirahmadi, M., *et al.*: 'NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC)', *IEEE Netw.*, 2014, **28**, (6), pp. 18–26
38 Courtney, M.: 'Stack them high [software-defined networking]', *IET Eng. Technol.*, 2014, **9**, (11), pp. 80–83
39 Riggio, R., Rasheed, T., Granelli, F.: 'EmPOWER: a testbed for network function virtualization research and experimentation'. Future Networks and Services (SDN4FNS), 2013, pp. 1–5
40 Batalle, J., Ferrer Riera, J., Escalona, E., *et al.*: 'On the implementation of NFV over an OpenFlow infrastructure: routing function virtualization'. Future Networks and Services (SDN4FNS), 2013, pp. 1–6
41 Monteleone, G., Paglierani, P.: 'Session border controller virtualization towards service-defined networks based on NFV and SDN'. Future Networks and Services (SDN4FNS), 2013, pp. 1–7
42 Cannistra, R., Carle, B., Johnson, R., *et al.*: 'Enabling autonomic provisioning in SDN cloud networks with NFV service chaining'. Optical Fiber Communication Conf., 2014, pp. 1–3
43 López, V., González de Dios, O., Fuentes, B., *et al.*: 'Towards a network operating system'. Optical Fiber Communication Conf., 2014, vol. 31, no. 6, pp. 1–3
44 Haleplidis, E., Salim, J.H., Denazis, S., Koufopavlou, O.: 'Towards a network abstraction model for SDN', *J. Netw. Syst. Manage.*, 2014, **23**, (7), pp. 1–19
45 Masutani, H., Nakajima, Y., Kinoshita, T., *et al.*: 'Requirements and design of flexible NFV network infrastructure node leveraging SDN/OpenFlow'. Optical Network Design and Modeling, 2014 Int. Conf. on, 2014, pp. 258–263
46 'OpenSDNCore'. Available at http://www.opensdncore.org/index.html, accessed February 2015
47 'Unify Project'. Available at http://www.fp7-unify.eu/, accessed February 2015
48 'T-Nova Project'. Available at http://www.t-nova.eu/, accessed February 2015
49 'METIS Project'. Available at https://www.metis2020.com, accessed February 2015
50 'CROWD Project'. Available at http://www.ict-crowd.eu, accessed February 2015
51 Osseiran, A., Boccardi, F., Braun, V., *et al.*: 'Scenarios for 5G mobile and wireless communications: the vision of the METIS project', *IEEE Commun. Mag.*, 2014, **52**, (5), pp. 26–35
52 Wang, J., Yan, Y., Dittmann, L.: 'Design of energy efficient optical networks with software enabled integrated control plane', *IET Netw.*, 2015, **14**, (1), pp. 30–36
53 'ICT-14–2014 Call', European Commission'. Available at http://www.ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/77-ict-14-2014.html, accessed February 2015

Pep Lluís Ferrer Gomila · M. Francisca Hinarejos Campos
(editores)

Actas de la XIV Reunión Española sobre Criptología y Seguridad de la Información

Actas de la XIV Reunión Española sobre Criptología y Seguridad de la Información

RECSI XIV

Maó, Menorca, Illes Balears, 26-28 Octubre de 2016

Publicado por:
Departamento de Ciencias Matemáticas e Informática
Universidad de les Illes Balears
Ctra. de Valldemossa, km 7.5. Palma (Illes Balears)
http://recsi16.uib.es

ISBN: 978-84-608-9470-4

Créditos:

Primera edición – Octubre 2016

Organizadores

Colaboradores

Seguridad en Redes de Nueva Generación: Arquitectura para la Gestión de Incidencias

Lorena Isabel Barona López, Jorge Maestre Vidal, Ángel Leonardo Valdivieso Caraguay,
Marco Antonio Sotelo Monge, Luis Javier García Villalba
Grupo de Análisis, Seguridad y Sistemas (GASS)
Departamento de Ingeniería del Software e Inteligencia Artificial (DISIA)
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, España
E-mail: {lorebaro, jmaestre, angevald, masotelo}@ucm.es, javiergv@fdi.ucm.es

***Resumen*—Las redes de nueva generación pretenden facilitar el despliegue de redes cada vez más rápidas y por las cuales fluya una mayor cantidad de información. Para cumplir con este objetivo, propone la combinación de diferentes tecnologías emergentes, tal es el caso de la auto-organización y virtualización de sus componentes. Esto a su vez plantea importantes desafíos en el campo de la seguridad. Con el fin de contribuir a su superación, el presente artículo introduce una arquitectura para la gestión de incidencias en redes móviles. La propuesta combina las bases de los esquemas de gestión de riesgos convencionales con el modelo de Consciencia Situacional propuesto por Endsley. Se ha tomado en cuenta diferentes aspectos, tales como la capacidad de adaptación a entornos de monitorización dinámicos, el seguimiento de contramedidas o la repercusión del contexto de la incidencia en el proceso de toma de decisiones. La propuesta cubre todos los niveles de tratamiento de la información en redes móviles, desde la infraestructura de red hasta el despliegue de los actuadores encargados de aplicar las contramedidas.**

***Palabras clave*—5G, Gestión de Incidencias, SDN/NFV, Seguridad de la Información.**

I. Introducción

La rápida proliferación del uso de dispositivos móviles ha puesto de manifiesto la incapacidad de las redes actuales para dar soporte a la inmensa cantidad de información que tendrán que gestionar [1]. Esta situación ha motivado el desarrollo de una nueva generación de redes móviles que no solo debe ser capaz de brindar la solución a este problema; también debe mejorar muchas de las características de sus predecesoras, tales como la tasa de transferencia de datos, interoperabilidad o consumo energético [2]. Alcanzar estos objetivos requiere de gran capacidad de innovación, la cual deberá reflejarse en cambios que abarquen desde el uso generalizado de la emisión de datos a Frecuencias Extremadamente Altas (EHF) hasta la manera en que se gestiona la información [3]. Esta última parte tiene un impacto especialmente relevante en los modelos de negocio basados en servicios y aplicaciones en tiempo real, donde tecnologías emergentes tales como Redes Definidas por Software (SDN) o Virtualización de Funciones de Red (NFV), han facilitado la personalización en los patrones y tratamiento del tráfico. Sin embargo, el desarrollo de estos servicios está limitado por la falta de estrategias eficientes de gestión y toma de decisiones [4], lo que conlleva mayor dificultad a la hora de desplegar medidas de seguridad.

En los entornos de monitorización actuales, la gestión de la seguridad de la información habitualmente se lleva a cabo mediante la aplicación de directivas o estándares que sirven de guía para proteger los recursos disponibles. Entre ellas se incluyen normas como ISO/IEC-27000 [5], NIST-SP800 [6], CVSS [7] o MAGERIT [8]; y plataformas como ITIL o COBIT [9]. Sin embargo sus bases han demostrado deficiencias al ser implementadas sobre escenarios especialmente dinámicos, donde el contexto juega un papel muy relevante a la hora de tomar decisiones [10]. Este es el caso de los entornos de monitorización basados en redes, y en especial, de aquellos que involucran tecnologías de nueva generación. Como solución a este problema algunos autores han adoptado metodologías de gestión de incidencias capaces de tratar la información de manera cognitiva, y que por lo tanto, facilitan su comprensión por medio del análisis contextual. De entre ellas destacan las que se basan en construir la Consciencia Situacional del entorno protegido mediante la aplicación del modelo de Ensdley, donde se estudia la percepción, comprensión y proyección del estado del sistema [11]. La adaptación de este paradigma a la gestión de la seguridad en redes ha llevado a acuñar el término Seguridad en Redes basada en Consciencia Situacional (NSSA) [12]. Sin embargo, a pesar de que se ha demostrado su eficacia en redes actuales, aún no ha sido adaptado a las dificultades que plantean las tecnologías 5G. Con el fin de contribuir a su desarrollo, en este artículo se introduce una arquitectura para la gestión de incidencias en redes de nueva generación. La propuesta combina las bases de los esquemas de gestión de riesgos tradicionales con el modelo de Consciencia Situacional publicado por Endsley. En ella se cubren todos los niveles de procesamiento de información de las redes 5G, desde su infraestructura hasta los actuadores encargados de aplicar las acciones de mitigación.

Este trabajo está estructurado en 5 secciones, siendo la primera de ellas la presente introducción. En la sección II se describen los trabajos relacionados. En la sección III se discuten las dificultades en la gestión de incidencias en redes 5G. En la sección IV se propone una arquitectura de seguridad para redes 5G. Finalmente en la sección V se presentan las conclusiones y el trabajo futuro.

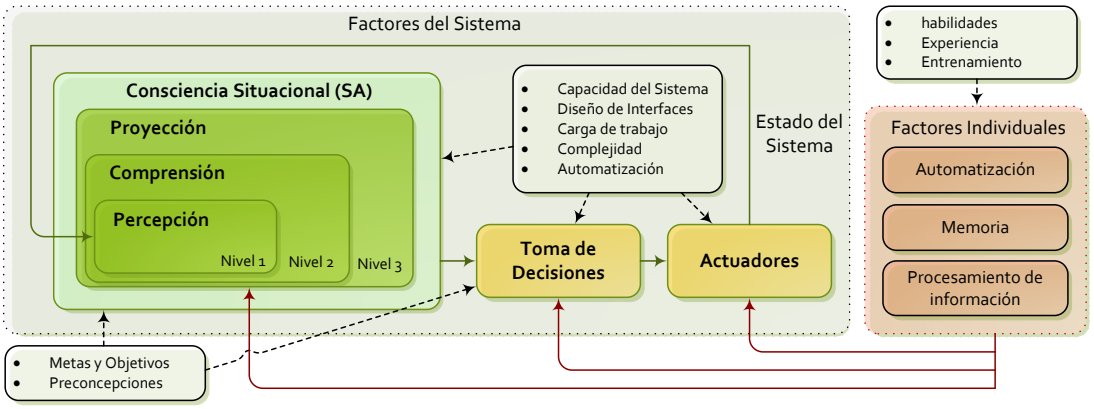

Figura 1. Consciencia Situacional según el modelo de Endsley

II. Trabajos Relacionados

II-A. Gestión de Riesgos

El problema de la gestión de riesgos es un tema de interés en la comunidad investigadora desde hace más de cuatro décadas. En consecuencia han sido publicados diversos trabajos que tratan de recopilar las contribuciones más relevantes, siendo [13], [14] algunas de las más actuales. En términos generales la bibliografía abarca una colección muy grande de tópicos que estudian desde la propia definición de riesgo y su planteamiento científico [15], [16], hasta el cómo son tratados a nivel gubernamental [17]. La necesidad de gestionar la defensa de las tecnologías de la información ha dado pie a diferentes herramientas para guiar a las organizaciones a su implementación, incluyendo estándares [5], [6], [8] y plataformas [7], [9]. La mayor parte de estas aproximaciones coinciden en que el proceso de gestión de incidencias debe recorrer las siguientes etapas: definición de riesgos, evaluación, monitorización y respuesta [14]. En la primera de ellas se lleva a cabo la delimitación de las situaciones observables en el sistema con características potencialmente dañinas, teniendo en cuenta los objetivos de las organizaciones a proteger, sus políticas, riesgos tolerables y principios de actuación [5]. A continuación se procede a la identificación de los posibles riesgos, su valoración y el planteamiento de métricas que permitan medir su impacto en el sistema [7], [8]. Por su complejidad esta es la fase con mayor presencia en la bibliografía [18], habiendo motivado el desarrollo de sistemas específicos para la Evaluación de Riesgos en la Seguridad de la Información (ISRA). En la etapa de monitorización se examina el entorno protegido en busca de indicios de riesgos. En el caso de detectarse alguno de ellos, tiene lugar la etapa de respuesta, en la que se lleva a cabo el despliegue de contramedidas. Debido a las grandes diferencias que existen entre los diferentes escenarios de monitorización, el éxito de la toma de decisiones depende directamente de los procesos anteriores y de su capacidad de adaptación a cada caso de uso. Los siguientes son ejemplos de metodologías para facilitar sus integración en ambientes más específicos: [19] para el control industrial, [20] para sistemas embebidos y [21] en SCADA.

II-B. Consciencia Situacional

Según Endsley, Consciencia Situacional (SA) significa «tener conocimiento del estado actual de un sistema, entender sus dinámicas, y ser capaces de predecir cambios» [11]. Su modelo se divide en tres etapas (ver Fig. 1): percepción, comprensión y proyección; en la primera se llevan a cabo las labores de monitorización e identificación de incidencias, en la segunda su asociación y en la tercera se predice la evolución del estado del sistema. A partir de dichas etapas se deciden las acciones a realizar y su modus operandi. Nótese que en este modelo existe realimentación entre los niveles de actuación/decisión con la SA, de tal manera que los resultados obtenidos influyen en las decisiones a tomar, facilitándose el uso de técnicas de diagnóstico avanzadas [12]. Este modelo ha sido implementado en diversas áreas, como gestión de incidencias en redes eléctricas inteligentes [24], generación de energía [25] o sistemas para evitar colisiones de vehículos [26]. En propuestas como [27] se plantea su adaptación a situaciones críticas por medio de distribución y priorización del tratamiento de datos. Tal y como se discute en [28], el modelo de Endsley ha demostrado ser efectivo en escenarios complejos y dinámicos, donde el diagnóstico tiene una alta dependencia del contexto en el que se han reportado las incidencias [10]. En la seguridad de la información también ha jugado un papel esencial [29], con un claro predominio de las implementaciones para la gestión de riesgos en situaciones de emergencia, sistemas industriales y redes. En ellas son mejoradas tres de las deficiencias de los sistemas para la Gestión de Riesgos en la Seguridad de la Información (ISRM) más repetidas en la bibliografía: no aprovechar todas las posibles fuentes de información, estimación de riesgos sin tener suficientemente en cuenta el contexto en que son registrados o su proyección, y la dificultad de llevar a cabo procesos de auditoría continuos [30]. Para tratar de dar solución a estos problemas sin perder la perspectiva aportada por los ISRM/ISRA, algunos trabajos han combinado ambos paradigmas, tal y como se observa en [10], [31], donde la Consciencia Situacional se construye teniendo en cuenta la definición de riesgos y su evaluación al aplicar directivas.

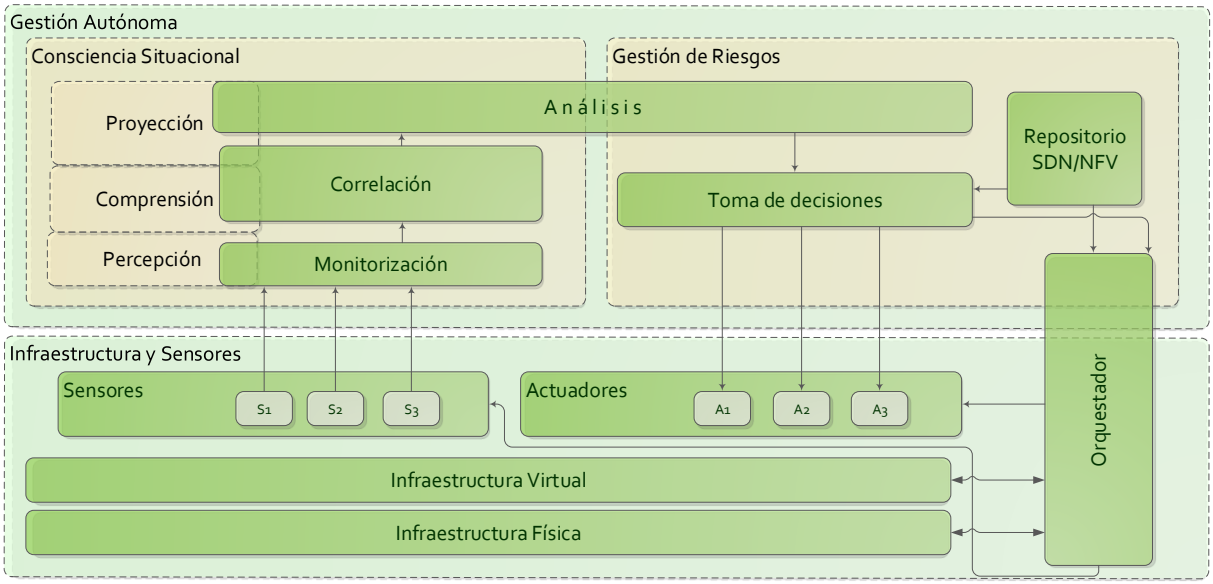

Figura 2. Arquitectura de gestión de riesgos para redes 5G

III. Gestión de Incidencias en 5G

Los principios de diseño de las nuevas plataformas móviles 5G tienen como objetivo el soporte de un incremento exponencial de dispositivos conectados y del tráfico que circula por la red. El soporte en tiempo real de estos servicios requiere, a diferencia de arquitecturas rígidas tradicionales, eliminar la estrecha unión hardware/software propietario y permitir una visión global junto con configuración y actualización dinámica de las diferentes operaciones de la red. Con este objetivo, modelos de diseño basados en tecnologías SDN/NFV, utilizadas inicialmente en redes de datos cableadas, han sido extendidos a plataformas inalámbricas y móviles. De esta manera, los operadores pueden gestionar la infraestructura móvil evitando la configuración manual, individual y remota de los diferentes equipos (generalmente utilizando línea de comandos CLI). Sin embargo, la gestión automática de riesgos en plataformas móviles que aprovechen este nuevo paradigma es escasa o prácticamente nula. El reto primordial es la coordinación entre los dispositivos repartidos en diferentes puntos de la infraestructura y las funciones virtuales que pueden ser instanciadas dinámicamente. De igual manera, dichos avances están limitados por la falta de esquemas que faciliten el procesamiento de altas cantidades de información que sirvan para detectar problemas en la red, y en la forma de analizar las causas de dichos problemas. Es claro que los diferentes eventos o incidencias tienen que ser organizados y priorizados adecuadamente de tal manera que no comprometan la seguridad de la información y la calidad de servicio que circula por la red.

IV. Arquitectura para la seguridad de la información en redes 5G

En la arquitectura propuesta, el análisis situacional tiene como objetivo la lectura del estado actual de los elementos monitorizados (infraestructura de red) y en caso de ser necesario, la respuesta automática a problemas de red identificados de manera reactiva o preventiva. Con este fin, se han identificado los siguientes requerimientos y supuestos:

- Los elementos de infraestructura monitorizados son compatibles con la tecnología SDN/NFV. En el caso de existir elementos de hardware/software no compatibles, se considera una capa de compatibilidad, la cual emula el funcionamiento de un elemento virtualizado compatible.
- La comunicación entre los diferentes elementos de la arquitectura se desarrollan por medio de canales seguros.
- La información proveniente de los elementos de monitorización (métricas, alertas) se consideran confiables.
- Los procesos de Análisis Situacional se ejecutan de manera independiente y transparente. Es decir, estas operaciones no afectan el rendimiento de los elementos de la infraestructura 5G.
- Los módulos funcionales representan tareas que en la práctica pueden ser implementados en arquitecturas distribuidas según las necesidades de velocidad de conexión y capacidad de cómputo.

En la Fig. 2 se ilustra la distribución de los diferentes módulos y la sinergia entre la arquitectura propuesta y el modelo de Endsley. Se han definido cuatro niveles lógicos principales: Infraestructura Virtual y Sensores, Monitorización/Correlación, Análisis y Decisión/Actuadores, donde el primero abarca las tareas relacionadas con la percepción, el segundo las de comprensión y el tercero las de decisión y ejecución de contramedidas. Por lo tanto Infraestructura/Sensores y Monitorización/Análisis construyen el SA, debiendo mantener una realimentación con Decisión/Actuadores que permita hacer el seguimiento de diagnóstico. A continuación se describe el rol de cada componente en la arquitectura.

IV-A. Infraestructura Virtual y Sensores

El principal objetivo de esta capa de procesamiento de datos es dar soporte al despliegue de los elementos necesarios para la captura de la información requerida para inferir riesgos del sistema. Su despliegue aprovecha uno de los principios de diseño de la nueva arquitectura móvil 5G: capacidad de integración con ambientes virtuales y la nube. De este modo se facilita el despliegue dinámico de elementos de red, promoviéndose el desarrollo de una capa de infraestructura completamente virtualizada: todos los elementos físicos o hardware tales como estaciones base, enlaces, encaminadores o servidores son gestionados por una capa de virtualización. A este nivel la tecnología SDN desacopla los planos de datos y de control de cada uno de los dispositivos de red. El usuario puede crear aplicaciones de software para modificar dinámicamente el comportamiento del plano de datos. Por su parte, NFV permite el tratamiento de las diferentes funciones de red (cortafuegos, DPI, balanceador) como funciones de software independientes del equipo. De esta manera, se consideran las aplicaciones tipo SDN (SDN-Apps) o NFV (NFV-Apps). En la arquitectura propuesta, los sensores son un tipo de NFV-Apps encargados de la monitorización de diferentes métricas del sistema. Ejemplos de sensores pueden ser: analizadores de tráfico, detectores de anomalías, monitores de la calidad de servicio (QoS/QoE), etc. Al ejecutarse sobre un ambiente virtualizado, los sensores (NFV-Apps) pueden cambiar dinámicamente su posición y características de monitorización. Esto permite aumentar la vigilancia en las regiones que están siendo atacadas, y delimitar zonas de cuarentena. Además facilita la realización de cambios en los parámetros a observar, dando pie a la posibilidad de considerar elementos de monitorización de propósito general adaptables a las circunstancias de la red.

IV-B. Monitorización y Correlación

La capa de monitorización recoge la información proveniente de los niveles inferiores (infraestructura virtual y sensores) y aplica técnicas de correlación para simplificar su análisis. Por lo tanto cuenta con dos componentes: extracción de datos y correlación. A continuación se describe cada uno de ellos:

- *Monitorización*. Los principales objetivos de la monitorización son recopilar y gestionar información proveniente de todas las fuentes de información, y facilitar su acceso a capas superiores. Este módulo también gestiona el registro y acceso de nuevos sensores. La información recabada es organizada en estructuras de datos eficientes tomando en cuenta la alta cantidad de información a procesar. En este sentido se han considerado dos escenarios: en el primero de ellos el sensor envía un reporte al monitor cuando encuentra información considerada importante (alertas, caída de un enlace, sobrecarga de memoria o CPU); en el otro escenario, cuando el monitor considere oportuno puede solicitar al sensor información necesaria para las tareas de agregación o análisis (topología virtual, enlaces libres, entre otros).
- *Correlación*. Se encarga del primer nivel de abstracción del procesamiento de información, en el cual, con el objetivo de tener una visión global del estado de la red, se ejecutan procesos de correlación y agregación. La información considerada redundante o no sensitiva es descartada. Es decir, por ejemplo, en el caso de recibir múltiples alertas provenientes de cada dispositivo de una misma zona afectada, se indica una única alerta junto con la topología afectada. Debido al dinamismo que ofrecen los ambientes virtuales en contraste con la rigidez de los elementos físicos, la topología se encuentra expresada como un grafo extendido o aumentado ($G_a(V_a, E_a)$), el cual modela los nodos (V_a) y enlaces (E_a) virtuales localizados en la infraestructura física [32], [33]. Asimismo, el resultado de operaciones de correlación y agregación permiten que las métricas recibidas a bajo nivel puedan ser expresadas o traducidas en métricas de alto nivel, también conocidas como estado de red (Health of Network HoN). Por ejemplo, la tasa de transmisión (Mbps), retardo (ms) y jitter (ms) de un flujo de datos de video *streaming*, recibido por los sensores en diferentes puntos de la red, puede ser expresado como una percepción global de la calidad de servicio QoS/QoE, cuantificada mediante la medición del MOS (Mean Opinion Score).

IV-C. Análisis

En el componente de análisis se lleva a cabo la identificación de situaciones de red a partir de las métricas recibidas desde el módulo de correlación y se construyen diagnósticos que facilitarán la decisión de las contramedidas a aplicar. En ella se distinguen dos tipos de situaciones: *eventos* y *riesgos*. Los *eventos* son incidencias que a priori no presentan naturaleza dañina, pero que sin embargo pueden mejorar el diagnóstico y la valoración de riesgos. Ejemplos de eventos son la detección de nuevos dispositivos de red, identificación de dispositivos inactivos o el despliegue de nuevas capas de virtualización. Por otro lado, los *riesgos* son situaciones que directamente conllevan vulneraciones en la seguridad del entorno protegido, como la explotación de vulnerabilidades, ataques de denegación de servicio o accesos no autorizados. Pueden ser inferidos a partir de eventos u otros riesgos. En Fig. 3 se ilustran las principales etapas del proceso de análisis: detección, identificación y evaluación de riesgos, gestión del inventario de activos, construcción de mapa de riesgos, predicción, diagnóstico y seguimiento de contramedidas. A continuación se describe brevemente cada una de ellas.

- *Detección*. Enlace entre los módulos de monitorización con las funciones de comprensión de la información. Tiene como datos de entrada las métricas de alto nivel construidas a partir de datos agregados y detecta las posibles situaciones inferibles a partir de ellos.
- *Identificación y evaluación de riesgos*. Implementación de normativas y/o plataformas para la identificación y evaluación de riesgos. Este componente puede formar parte de las funciones del módulo de detección o actuar de manera independiente. Dada la taxonomía propuesta en [14] y las características de 5G, es recomendable que se consideren criterios de evaluación cualitativos desde

una perspectiva basada en servicios, teniendo en cuenta la propagación de las incidencias a lo largo de la red.

- *Inventario de activos*. Las redes 5G tienen la capacidad de automatizar el despliegue de nuevos servicios y dispositivos de red en función de su estado, situación que implica gran dificultad a la hora de medir el impacto de los riesgos. Con el fin de contribuir a su desarrollo, este módulo se encarga de la construcción y mantenimiento del inventario de activos del sistema.
- *Mapa de riesgos*. Con el fin de facilitar las tareas de diagnóstico y toma de decisiones, en este componente se genera y gestiona un mapa de riesgos de la red. En su construcción participan las métricas correlacionadas ofrecidas por las capas de percepción y las situaciones detectadas a nivel de análisis.
- *Predicción*. Proyección de la Consciencia Situacional en el modelo de Endsley. A partir del mapa de riesgos y las incidencias identificadas, este componente aplica algoritmos para anticipar cambios en la red.
- *Diagnóstico*. Análisis de alto nivel de riesgos, su valoración, impacto, proyección y estado de la red con el fin de identificar situaciones de mayor complejidad poco visibles en los niveles inferiores. Por ejemplo, el componente de diagnóstico puede reconocer botnets por medio de la relación de riesgos que determinan la presencia de dispositivos de red infectados y tráfico anómalo en sus proximidades.
- *Seguimiento de Contramedidas*. Principal enlace entre las tareas de análisis con las de toma de decisiones. El sistema de seguimiento de contramedidas comunica los problemas diagnosticados a las siguientes capas de procesamiento de información y construye un historial con las acciones ejecutadas para su mitigación. Según el modelo de Endsley, forma parte de la realimentación entre actuadores y comprensión. Además mejora las tareas de diagnóstico, facilita la realización de cambios en las decisiones a tomar en función de los resultados de situaciones previas similares.

IV-D. Toma de decisiones y Actuadores

La toma de decisiones busca mitigar los problemas que afectan la normal operación de los elementos de la red y, de ser el caso, optimizar el rendimiento de los diferentes servicios que se brindan. Con este objetivo, el sistema recibe la información proveniente de la etapa de análisis y selecciona un conjunto de acciones o respuestas a ejecutarse. Las acciones disponibles se encuentran repartidas entre las diferentes funciones NFV-Apps. Por ejemplo, en respuesta a un ataque de denegación de servicio, el sistema utiliza el informe recibido por análisis y toma la decisión de instalar funciones de firewall en puntos estratégicos. De igual manera, la información proveniente de análisis puede incluir resultados de algoritmos de predicción. Estos datos sirven para ejecutar acciones de manera proactiva, es decir, evitar o disminuir la probabilidad de que los servicios se vean afectados negativamente. Cuando el número de usuarios conectados a un servicio se incrementa paulatinamente y

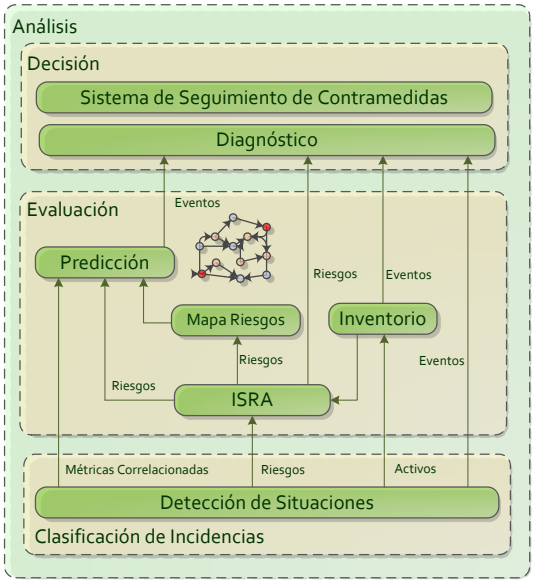

Figura 3. Análisis de situaciones en 5G

el análisis determina que el tráfico es legítimo y pronostica un incremento de tráfico, el sistema puede automáticamente instanciar balanceadores de carga para evitar un futuro colapso del servicio. La ejecución de las diferentes acciones es coordinada por un agente orquestador, el cual se asegura que los recursos virtuales para aplicar dichas acciones se encuentran disponibles y no afectarán el rendimiento del sistema.

V. Conclusiones

Los esquemas para la ISRM convencionales han mostrado importantes carencias en su despliegue sobre escenarios de monitorización dinámicos, tal y como sucede cuando actúan sobre redes 5G. Con el fin de contribuir a su desarrollo, en este artículo se ha presentado una arquitectura para la gestión de incidencias en 5G basada en la combinación del modelo cognitivo de Consciencia Situacional propuesto por Endsley, y las plataformas y normativas más frecuentes en la identificación y evaluación de incidencias. Esto facilita la automatización de medidas proactivas/reactivas para hacer frente a incidencias, y mejora aspectos como el aprovechamiento de todas las fuentes de información en los ISRM, la calidad del contexto a tener en cuenta a la hora de tomar decisiones y la proyección del estado del sistema. La arquitectura propuesta es factible gracias a las ventajas que presentan tecnologías innovadoras, tales como SDN, NFV, virtualización, análisis y predicción, entre otros. El diseño reduce de manera significativa el gasto de capital y operacional. Además permite el tratamiento de información de redes 5G mediante métricas de alto nivel (HoN).

Con el fin de no añadir complejidad adicional a esta primera aproximación, algunos aspectos a tener en cuenta de cara a su implementación no han sido desarrollados en detalle. Este es el caso de las características de los sensores/actuadores y su relación con los repositorios a partir de los cuales pueden ser instanciados. No se ha profundizado en los métodos de

diagnóstico avanzados para aprovechar el seguimiento de contramedidas ni en su realimentación con la toma de decisiones Tampoco en la especificación de las APIs e interfaces que comunican los distintos componentes, o en el lenguaje táctico a partir del que comparten información; todos estos aspectos quedarán cubiertos en trabajos futuros.

A partir de la arquitectura propuesta han surgido diferentes líneas de investigación. Las más evidentes se centran en su implementación sobre entornos de monitorización reales, como es el caso del proyecto Europeo que financia el estudio realizado. Otras se centran en el análisis de sus diferentes casos de uso, extendiéndose el concepto de incidencia a diversos problemas relacionados con la calidad de experiencia del usuario (QoE); por ejemplo, la solución de errores internos y colisiones en red, o la optimización en la transferencia de contenido multimedia. Finalmente cabe destacar el interés que esta arquitectura ha suscitado como alternativa a muchos de los esquemas defensivos colaborativos actuales, y su gran potencial a la hora de sincronizar los esfuerzos de prevención, detección, mitigación e identificación del origen de ataques complejos.

Agradecimientos

Los autores agradecen la financiación que les brinda el Programa Marco de Investigación e Innovación Horizonte 2020 de la Comisión Europea a través del Proyecto H2020-ICT-2014-2671672-SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks). Ángel Leonardo Valdivieso Caraguay y Lorena Isabel Barona López son auspiciados por la Secretaría Nacional de Educación Superior, Ciencia y Tecnología e Innovación SENESCYT (Quito-Ecuador).

Referencias

[1] J.G. Andrews, S. Buzzi, W. Choi, S.V. Hanly, "What Will 5G Be?," *Journal on Selected Areas in Communications*, vol. 32 (6), pp. 1065–1082, 2014.

[2] N. Panwar, S. Sharma, A.K. Singh, "A survey on 5G: The Next Generation of Mobile Communication," *Physical Communication*, vol. 18 (2), pp. 64–84, 2016.

[3] F. Boccardi, R.W.Heath, A. Lozano, T.L. Marzetta, P. Popovski, "Five Disruptive Technology Directions for 5G," *IEEE Communications Magazine*, vol. 52 (2), pp. 74–80, 2014.

[4] A. Imran, A. Zoha, A. Abu-Dayya, "Challenges in 5G: How to Empower SON with Big Data for Enabling 5G," *IEEE Network*, vol. 28 (6), pp. 27–33, 2014.

[5] ISO/IEC 27002 series, "Information Technology - Security Techniques - Code of Practice for Information Security Management," (2016). Disponible en http://www.iso.org.

[6] NIST-SP800 series, Special Publications on Computer Security (2016). Disponible en http://http://csrc.nist.gov/publications/PubsSPs.html.

[7] CVSS: Common Vulnerability Scoring System, Special Publications on Computer Security (2016). Disponible en https://www.first.org/cvss/specification-document.

[8] MAGERIT: Risk Analysis and Management Methodology for Information Systems, (2016). Disponible en https://http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/pae_Magerit.html.

[9] R. Parvizi, F. Oghbaei, S. R. Khayami, "Using COBIT and ITIL Frameworks to Establish the Alignment of Business and IT Organizations as one of the Critical Success Factors in ERP Implementation," en *Proc. 5th Conference on Information and Knowledge Technology (IKT)*, 2013, pp. 274–278.

[10] J. Webb, A. Ahmad, S.B. Maynard, G. Shanks, "A Situation Awareness Model for Information Security Risk Management," *Computers & Security*, vol. 44, pp. 1–15, 2014.

[11] M.R. Endsley, "Design and Evaluation for Situation Awareness Enhancement," en *Proc. 32nd Annual Meeting on Human Factors Society*, 1988, pp. 97–101.

[12] Y.B. Leau, S. Manickam, "Network Security Situation Prediction: A Review and Discussion," Intelligence in the Era of Big Data,*Communications in Computer and Information Science*, vol. 516, pp. 424–435, 2015.

[13] T. Aven, "Risk Assessment and Risk Management: Review of Recent Advances on their Foundation," *European Journal of Operational Research*, vol. 253 (1), pp. 1-13, 2016.

[14] A. Shameli-Sendi, R. Aghababaei-Barzegar, M. Cheriet, "Taxonomy of Information Security Risk Assessment (ISRA)," *Computers & Security*, vol. 57, pp. 14-30, 2016.

[15] S.O.Hansson, T. Aven, "Is Risk Analysis Scientific?," *Risk Analysis*, vol. 34 (7), pp. 1173–1183, 2014.

[16] E. Hollnagel, "Is Safety a Subject for Science?," *Safety Science*, vol. 67, pp. 21–24, 2014.

[17] P. Doty, "U.S. Homeland Security and Risk Assessment,"*Government Information Quarterly*, vol. 32 (3), pp. 342–352, 2015.

[18] M. Yang, F. Khan, L. Lye, P. Amyotte, "Risk Assessment of Rare Events," *Process Safety and Environmental Protection*, vol. 98, pp. 102–108, 2015.

[19] W. Knowles, D. Prince, D. Hutchison, K. Jones, "A Survey of Cyber Security Management in Industrial Control Systems," *International Journal of Critical Infrastructure Protection*, vol. 9, pp. 52–80, 2015.

[20] S. Ni, Y. Zhuang, J. Gu, Y. Huo, "A Formal Model and Risk Assessment Method for Security-critical Real-time Embedded Systems," *Computers & Security*, vol. 58, pp. 199–215, 2016.

[21] Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, K. Stoddart, "A Review of Cyber Security Risk Assessment Methods for SCADA Systems," *Computers & Security*, vol. 56, pp. 1–27, 2016.

[22] M.R. Endsley, "A Comparative Analysis of SAGAT and SART for Evaluations of Situation Awareness," en *Proc. 42nd Annual Meeting on Human Factors and Ergonomics Society*, 1988, pp. 82–86.

[23] E.C. Adam, "Fighter Cockpits of the Future," en *Proc. 12th IEEE/AIAA Digital Avionics Systems Conference (DASC)*, 1993, pp. 318–323.

[24] N. Dahal, O. Abuomar, R. King, V. Madani, "Event Stream Processing for Improved Situational Awareness in the Smart Grid," *Expert Systems with Applications*, vol. 42(20), pp. 6853–6863, 2015.

[25] M. Naderpour, S. Nazir, J. Lu, "The Role of Situation Awareness in Accidents of Large-scale Technological Systems," *Process Safety and Environmental Protection*, vol. 97, pp. 13–24, 2015.

[26] E. Moradi-Pari, A. Tahmasbi-Sarvestani, Y.P. Fallah, "A Hybrid Systems Approach to Modeling Real-Time Situation-Awareness Component of Networked Crash Avoidance Systems," *IEEE Systems Journal*, vol. 10(1), pp. 169–178, 2016.

[27] H. Seppänen, K. Virrantaus, "Shared Situational Awareness and Information Quality in Disaster Management," *Safety Science*, vol. 77, pp. 112–122, 2015.

[28] M.M. Chatzimichailidou, N.A. Stanton, I.M. Dokas, "The Concept of Risk Situation Awareness Provision: Towards a New Approach for Assessing the DSA about the Threats and Vulnerabilities of Complex Socio-technical Systems," *Safety Science*, vol. 79, pp. 126–138, 2015.

[29] U. Franke, J. Brynielsson, "Cyber Situational Awareness - A Systematic Review of the Literature," *Computers & Security*, vol. 46, pp. 18–31, 2014.

[30] R. Schmittling, A. Munns, "Performing a Security Risk Assessment," *ISACA Journal*, vol. 1, pp. 1–7, 2010.

[31] M. Naderpour, J. Lu, G. Zhang, "A Situation Risk Awareness Approach for Process Systems Safety," *Safety Science*, vol. 64, 173–189, 2014.

[32] S. Shanbhag, A. Kandoor, C. Wang, "VHub: Single-stage Virtual Network Mapping Through Hub Location," *Computer Networks*, vol. 77, 169–180, 2015.

[33] N. Chowdhury, M. Rahman, R. Boutaba, "Virtual Network Embedding with Coordinated Node and Link Mapping," *INFOCOM*, 2009, pp. 783–791.

future internet

MDPI

Article

Key Technologies in the Context of Future Networks: Operational and Management Requirements

Lorena Isabel Barona López †, Ángel Leonardo Valdivieso Caraguay †, Marco Antonio Sotelo Monge † and Luis Javier García Villalba *,†

Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), Faculty of Computer Science and Engineering, Office 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases , 9, Ciudad Universitaria, 28040 Madrid, Spain; lorebaro@ucm.es (L.I.B.L.); angevald@ucm.es (Á.L.V.C.); masotelo@ucm.es (M.A.S.M.)

* Correspondence: javiergv@fdi.ucm.es; Tel.: +34-91-394-7638

† These authors contributed equally to this work.

Academic Editor: Dino Giuli

Received: 11 September 2016; Accepted: 7 December 2016; Published: 22 December 2016

Abstract: The concept of Future Networks is based on the premise that current infrastructures require enhanced control, service customization, self-organization and self-management capabilities to meet the new needs in a connected society, especially of mobile users. In order to provide a high-performance mobile system, three main fields must be improved: radio, network, and operation and management. In particular, operation and management capabilities are intended to enable business agility and operational sustainability, where the addition of new services does not imply an excessive increase in capital or operational expenditures. In this context, a set of key-enabled technologies have emerged in order to aid in this field. Concepts such as Software Defined Network (SDN), Network Function Virtualization (NFV) and Self-Organized Networks (SON) are pushing traditional systems towards the next 5G network generation.This paper presents an overview of the current status of these promising technologies and ongoing works to fulfill the operational and management requirements of mobile infrastructures. This work also details the use cases and the challenges, taking into account not only SDN, NFV, cloud computing and SON but also other paradigms.

Keywords: 5G; future network; NFV; SDN

1. Introduction

The emergence of a new business model and services (e-solutions, e-health, e-commerce, Voz IP, streaming, among others) and the exponential growth in the information circulating on the Internet has brought unexpected challenges to the IT industry. The development of new infrastructures, known as Future Networks, is focused on ensuring robustness, security, scalability and the fast deployment of applications through the customization of network behavior.

According to the Future Internet 2020 Report of the European Commission, the development of a new generation of networks takes an average of 10 years, this means that the concept of Future Networks is coming soon. Future Networks must provide a flexible, reliable, secure, smart and high-performance environment to connect the digital society, while leveraging the competitiveness, faster innovation and standardization of new technologies. This network must embrace not only current services but also any kind of elements (Internet of Things—IoT) [1]. These kind of networks will generate a significant impact not only on the societal but also on the operational field. On one hand, Future Networks must cover the necessities of smart cities, entertainment, public security, etc., providing a wide range of network services and applications [2]. Users will expect

enhanced Quality of Experience (QoE) with minimal disruptions of the services, regardless of their location, the kind of device, or when the service is required. On the other hand, Future Networks will help to decrease the capital and operational expenditures (capex/opex) related to the deployment and management of new applications and infrastructures with substantially reduced service creation time [3]. Nowadays, the introduction of novel technologies is a time-consuming process due the slow standardization process, manual service deployment or the semi-automated management tasks. In the context of mobile networks, the average revenue per user (ARPU) is continuously decreasing, while the demand on mobile traffic keeps growing. This causes a negative response by network operators to invest in new network hardware infrastructure. In order to lay the foundations of Future Mobile Networks, three fields must be improved: Radio, Network and Operations and Management capabilities [4].

- Radio capabilities leverage the spectrum optimization, enhance interference coordination mechanisms and support dynamic radio topologies through the exploitation of higher frequencies, enabling cost-effective dense deployments, intelligent and dynamic coordination of multi Radio Access Technology (RAT), as well as sharing resources, among others.
- Network functionalities will enable the creation of an open environment in order to support several use cases in a cost-effective manner by means of the enhancement of user devices, minimizing the number of deployed entities and splitting the control and user plane functions (open its interfaces).
- The operation and management capabilities are intended to simplify operations not only in network control tasks but also in the deployment of new services, without increasing the system complexity. This field also includes reactive and proactive mechanisms to enhance the decision-making in control and management operations. This characteristic will enable the deployment of virtualized components, wherever they might be needed.

Radio and Network capabilities are topics well-studied in the literature [5–8]. In [6], a detailed survey and ongoing projects related to 5G networks are presented. This work discusses some emerging technologies, such as massive Multiple Input Multiple Output (MIMO), cognitive radio, cloud technologies and Device to Device Communication (D2D) in order to tackle the following requirements: enhanced data rate, spectral efficiency, lower latency, deployment and management of ultra dense networks. For their part, Boccardi et al. [5] describe five disruptive concepts that might impact on the development of 5G Radio requirements. They take into account the ability for devices to communicate between themselves (Machine to Machine communication—M2M), spectrum and resource optimization (massive MIMO and millimeter wave), the introduction of a device-centric concept and smarter devices (allowed to play an active role in the network). Regarding Network capabilities, one of the main challenges is to create an open, multi-tenant and service-oriented environment to support large amounts of traffic while covering different kinds of Quality of Service (QoS) levels and Service Level Agreements (SLA), in terms of latency, bandwidth or jitter. This environment will allow a flexible reconfiguration of network devices and programmability features based on the device-level, application, user and environment context [7]. Meanwhile, the introduction of intelligence in 5G systems might enable the improvement of the resource use (spectrum, transmission power levels and other radio resources), cost-effective energy mechanisms and flexible cell management (different sizes) [9].

In order to tackle operation and management capabilities and enable ubiquitous connectivity, the research community proposes the introduction of some key technologies, such as SDN [10], NFV [11], Cloud Computing [12], Self-Organized Network (SON) [13] and Machine Learning [14]. SDN is based on the separation of the control plane from the data plane in traditional network devices. This decomposition allows the centralized control of the network with greater automation capacities and it simplifies the management process. For its part, NFV allows the implementation of traditional Network Functions (NF) as virtualized instances, running in a generic hardware. The main advantage

of NFV is its improved scalability capacity which, due to Virtual Network Functions (VNF), can be deployed anytime and anywhere in minutes, whereas previously it took more time (compared with traditional functions) [15]. From the technical point of view, SDN and NFV are complementary technologies, and together could facilitate configuration and network customization [16]. For their part, concepts such as Cloud Computing and SON allow the easy deployment of services (on-demand fashion) and enhanced traffic management based on intelligence decisions.

SDN, NFV, Cloud computing and SON are enablers that provide business agility and simplify the operation and management tasks. In contrast to traditional mobile systems, future networks will enable operators to control the traffic information (via SDN) in order to use only necessary network functions in a shared virtualized network (NFV and cloud computing). These technologies also allow the reduction of the complexity of planning, configuration and optimization tasks in the whole system, giving the capability to reuse existing infrastructures in a proactive way.

The main objective of this paper is to present a full view of the applicability of these technologies as well as the finished and ongoing projects in order to cover the operational and management requirements of the future mobile networks. It also provides the current use cases that will leverage the development of services. This paper is organized as follows. Section 2 presents the key-enabled technologies that could be taken into account for the design of future mobile networks. Section 3 presents an overview of the main research, projects and use cases that are based on these technologies. Then, Section 4 describes the challenges and future trends related to the adoption of these technologies and opens a discussion about the future of mobile networks. Finally, the conclusions of this work are presented in Section 5.

2. Key-Enabled Technologies for Future Networks

Future Networks envisage a fully connected society where the user can enjoy enhanced services and the operators obtain enough revenues from the deployment and provision of their services. In this context, Future Networks could benefit from the evolution of novel technologies such as SDN, NFV, SON and Cloud Computing, as is detailed below.

2.1. Software Defined Networking

Software Defined Networking [10,17] is a novel architecture in network communications. SDN proposes the separation between data and control planes in network devices and a centralized control of the network. In SDN, three well defined layers are established: infrastructure layer, control layer and application layer Figure 1.

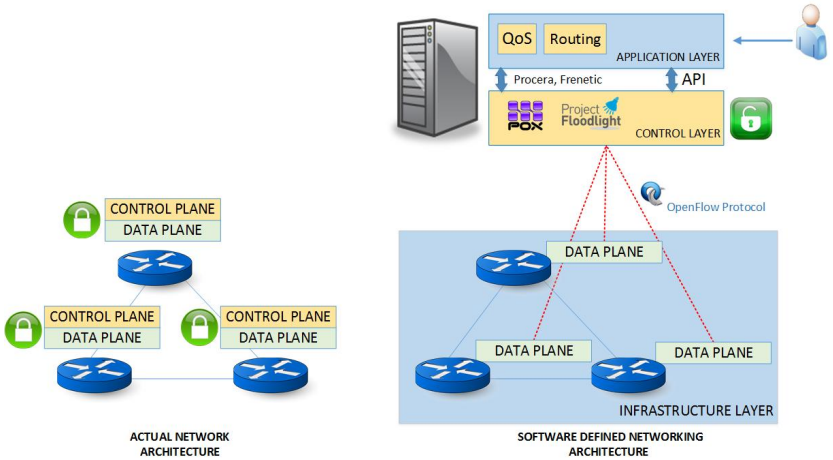

Figure 1. SDN vs. Traditional Architecture.

Infrastructure layer (data plane) refers to the physical hardware and/or basic software components of the network devices responsible for sending the packets through the network. In other words, this component does not make logical decisions about the network behavior. For its part, the control plane analyzes the information received by the data plane and makes decisions about the network behavior. The control plane offers functionalities to the application layer, which uses these functions to establish high level network policies or services. To connect hardware devices with a SDN controller, a standardized protocol is needed. In practice, OpenFlow is the first SDN Open Protocol initiative adopted by the research community [18]. OpenFlow uses the most common hardware capabilities of network devices (flow tables) and opens those up. With this approach, the whole network behavior can be controlled and dynamically adapted according to user needs. Figure 1 shows the principal differences between traditional and the SDN architectures and their components. It is important to note that SDN also eliminates the complexity and the closed nature of traditional networks. Future Networks can take advantage of the main characteristics of SDN in order to enhance the control, management and customization of the network services. In the context of mobile networks, SDN will allow the management of mobile resources in an easy manner, providing a better QoS in an open environment.

2.2. Network Function Virtualization

The concept of virtualization has allowed resource sharing (e.g., hardware) between different tenants, each one with its private functionality (e.g., different Operating System). For its part, network virtualization enables the sharing of network resources with different virtual topologies and forwarding logic in the same network infrastructure (e.g., Virtual Local Area Network-VLAN) [15]. However, if a service provider needs a specialized network application (firewalls, Deep Packet Inspection-DPI, etc), it must add a new hardware device. It represents an extra investment (capex/opex) and creates scalability and innovation constraints. In this context, the NFV has gained importance between network operators for its facilities to deploy services or applications as software network functions that can be automatically instantiated in different parts of the infrastructure. In 2012, main telecommunication service providers and the European Telecommunications Standards Institute (ETSI) proposed the NFV approach [11] . This allows the easy deployment of network functions on standard switches, storage or high volume servers as shown in Figure 2.

The referential architecture defines three main components: the Network Function Virtualization Infrastructure (NFVI); VNFs; and NFV Management & Orchestration (NFV M&O). NFVI controls the hardware resources (computing, storage and network), and uses a virtualization process in order to create software instances (virtual compute, virtual storage and virtual network). VNFs are the network functions that run over the NFVI. NFV M&O orchestrates and manages VNFs and the NFVI. Additionally, this module works with the external Business Support System and Operation Support System (OSS/BSS) in order to bill the services. The main objective of this model is to provide an architecture that allows the easy deployment, orchestration and management of traditional network functions as virtualized instances. These characteristics will enable future infrastructures to provide services regardless of the type of devices or their location and will promote on-demand service modality.

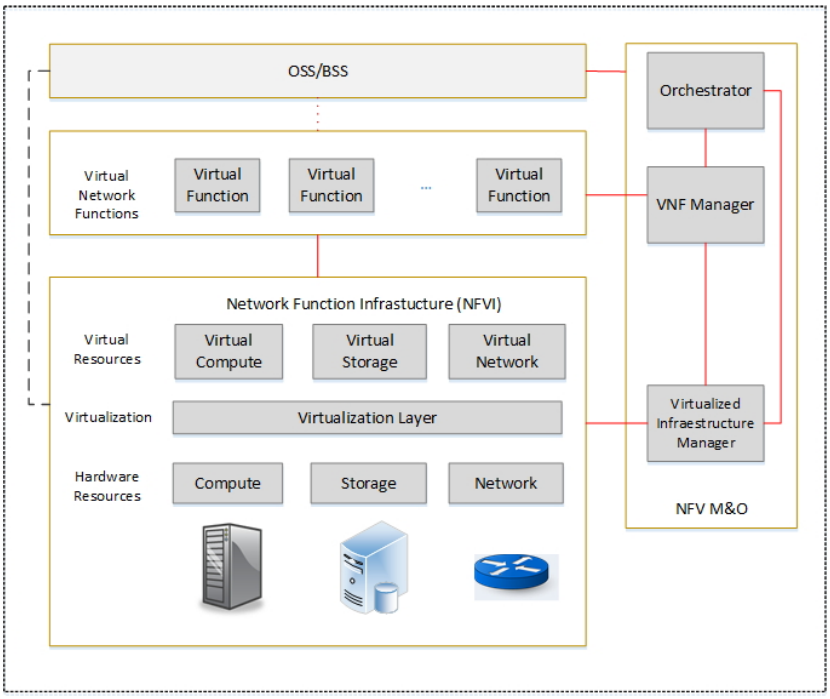

Figure 2. NFV Reference Architecture.

2.3. Cloud Computing

With the rapid evolution of the services, processes and enterprise business models, traditional provisioning of resources (computing, storage and network) has evolved to a new concept known as Cloud Computing [12]. In cloud computing, a particular resource can be leased by users to a third party, according to their needs. This innovative service-oriented model provides an elastic infrastructure, while decreasing the costs of hardware acquisition. Cloud computing offers different service models depending on the available resources to be leased. For instance, Software as a Service (SaaS) enables the sharing of software programs running on a cloud infrastructure. For its part, Platform as a Service (PaaS) architectures allow users to execute customized programs created with different libraries, tools, or programming languages supported by the cloud environment. In the Infrastructure as a Service (IaaS) model, the user is able to customize the different computing resources (processing, storage, and network) and deploy arbitrary software (e.g., different Operating Systems). In particular, Openstack [19] appears as a promising open source project to manage cloud platforms through its set of services (nova, neutron, keystone, telemetry, etc.), which could be integrated not only into traditional networks, but also SDN and NFV approaches. In the context of Future Networks, cloud computing would facilitate on-demand network access to available resources by means of its elastic capacity. This modality benefits the final users as well as the service provider.

2.4. Self-Organized Networks

The exponential increase of on-line services (e-bank, e-health, streaming) and the number of connected devices has brought new challenges to the network infrastructure in terms of security, performance and reliability. The management and rapid response to unexpected problems in the network (link failure, congestion, Distributed Denial of Service-DDoS, delay) is fundamental to guarantee QoS/QoE to users. Network intelligence mechanisms are needed in order to resolve/mitigate possible problems, to decrease the service recovery time and the operational costs [14]. Moreover, the use of advanced techniques such as artificial intelligence, data mining or pattern recognition enables proactive and reactive self-management actions capable of preventing potential problems and maintaining the subscribed network services. However, the implementation of

self-organized solutions [13] in current networks is limited by the rigidity of the traditional network architectures. The modification or customization of a flow in the network requires the individual configuration of each network device, and the deployment of new network services/protocols, from design to implementation, can take a long time. In this context, the smart integration of the novel technologies listed above has the potential to provide operators a smart network infrastructure capable of managing complex network scenarios and reducing operational costs, as shown in Figure 3.

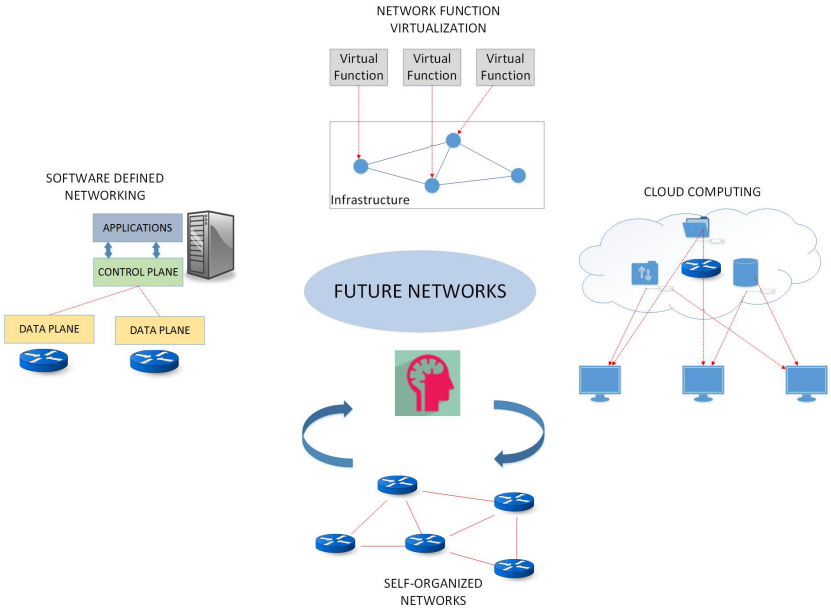

Figure 3. Trends in Future Networks.

The combination of SDN with NFV aims to enhance the management and orchestration process in current networks. On one hand, SDN can control the network behavior and it may require specific network functions in order to fulfill the new requirements and ensure the QoS levels established by the service provider. On the other hand, NFV allows the rapid deployment of these functions, without constraints of location or hardware vendor. For its part, cloud computing and SON could facilitate the scalability of the services, the customization of network infrastructure and the smart control and management of the available resources. It is expected that machine learning and data mining can provide higher and more intelligent mechanisms related to monitoring and management tasks [14]. It is important to note that future mobile networks envisage to provide a system driven by software, relying on technologies such as SDN, NFV, cloud computing and SON. These concepts aim to cover flexibility needs in terms of resource allocation, smarter monitoring and the introduction of new network functions, while ensuring the quality of the services. Furthermore, these technologies have been applied in radio and network fields in order to tackle their requirements.

3. Research Overview

Future Mobile Networks require customizable, efficient and scalable network infrastructures in order to meet the new user needs and the exponentially-increasing traffic demands, while decreasing the capital and operational expenditures. The SDN concept has been introduced in a broad range of fields, such as QoS, data centers, mobile and optical networks, security, network virtualization, among others [17]. As an instance, Google was one of the first enterprises to incorporate the SDN concept to communicate their internal Datacenter-WAN. Furthermore, there are some projects that allow SDN experimentation by offering scalable testbed infrastructures with research purposes, such as Geant, GENI, Ofelia, Felix, among others [20].

In particular, the integration of SDN or NFV with mobile networks includes the deployment of virtualized base stations and core components (Long Term Evolution-LTE) [21], energy efficiency experimentation on WiFi networks, the optimization of very dense and heterogeneous wireless networks [22], etc. The next generation of mobile networks could take advantage of the combination of key-enabled technologies to enhance the following areas: (i) the development of radio access (high speed, spectrum efficiency, high mobility, high availability); (ii) improvements in core networks (QoS support, aggregated processes, network slicing, cloud deployment) and (iii) the management and orchestration process (customization of user needs, dynamic allocation of resources, energy efficiency mechanism, manage a big amount of data) [5,9,23].

Different standard organizations leverage the adoption of SDN and NFV concepts in their infrastructures. These organizations have presented the challenges, Key Performance Indicators (KPI) and possible use cases in order to cover the above-mentioned areas. As an instance, Open Networking Foundation (ONF) [24] promotes the adoption of SDN and defines a wide range of use cases, such as inter-cell interference management, virtual customer edge, network virtualization or data center optimization. Meanwhile, NFV is an initiative of ETSI and telecommunication providers, which proposes the virtualization of the traditional network functions. ETSI-NFV defines nine general use cases [25], such as NFV Infrastructure as a Service (NFVIaaS), VNF Forwarding Graph (VNF-FG), etc. In the scope of mobile networks, NFV promotes the virtualization of Mobile Core Networks and IP Multimedia Subsystem (IMS), the virtualization of a mobile base station, the virtualization of the home environment and the virtualization of Content Delivery Networks (CDNs). In the meantime, some open source projects led by the research community have emerged to provide an open environment to test with SDN, NFV and cloud computing, such as OpenNFV (SDN and NFV) [26], Floodlight (OpenFlow and OpenStack support) [27], OpenDaylight (SDN, NFV and OpenStack) [28], among others.

Regarding mobile networks, industry manufacturers, telecommunication operators, and related stakeholders are working on the definition of requirements, standardization, regulation and development of future mobile systems, such as 5G-PPP (5G Infrastructure Public Private Partnership) and Next Generation Mobile Network initiative (NGMN). The 5G-PPP [29] proposes solutions, standards and infrastructures to allow the ubiquitous 5G communication. For its part, the NGMN [4] will expect to provide 5G solutions by 2020, within eight general use cases: broadband access in dense areas, broadband access everywhere, high user mobility, massive Internet of Things, extreme real-time communication, lifeline communication, ultra-reliable communication and broadcast-like services.

The most outstanding efforts have been made in the 5G research field. A wide range of projects or initiatives will expect to cover the needs of future mobile users. These worldwide initiatives encompass global regions of Asia, Europe and the Americas.

With the aim to promote the adoption of 5G in Asia, China has launched the IMT-2020 promotion group [30], which manages five working groups: Requirements, Technology, Spectrum, Intellectual Property Right (IPR) and Standardization. This is the most important promotion platform related with research and international cooperation purposes. Similarly, coordinated efforts in the 5G area have been launched in South Korea and Japan, the former with the 5G Forum [31] and the latter with the Fifth Generation Mobile Communications Promotion Forum (5GMF) [32]. Both are conducting research projects involving active participants from the government, industry, and academia, in order to facilitate the development of 5G.

Significant efforts have been made in Europe under the support of the European Union Framework Project 7 (FP7) and the Horizon 2020 programmes [29]. On the one hand, FP7 has launched 5G research projects such as METIS, MCN, CONTENT, T-NOVA, UNIFY, CROWD, etc. On the other hand, Horizon 2020 has financed several research projects (considered by 5G PPP as Phase 1 Projects) such as 5G-NORMA, METIS II, CHARISMA, SONATA, FLEX5GWARE, SELFNET, among others.

In the FP7 context, Mobile and wireless communications Enablers for the Twenty-twenty Information Society Project (METIS) [33] lays the foundations of 5G networks and promotes the general agreement to design this mobile environment. The first phase of this project (METIS I) includes five big

scenarios (amazingly fast, great Services in a crowd, ubiquitous things communicating, best experience follows you, super real-time and reliable connections) in a use-case driven approach.

The initiative Mobile Cloud Computing (MCN) [34] provides mobile services by means of the combination of three components: mobile network, compute, and storage resources. MCN defines a wide range of use cases, such as Radio Access Network (RAN) on Demand, Mobile Virtual Resources on Demand, Machine Type Communication on Demand, SDN or virtualized Evolved Packet Core (EPC), to mention a few. In the same way, the CONTENT project [35] proposes a network infrastructure that enables end-to-end cloud and mobile services. This project provides a virtualized infrastructure based on LTE, WIFI and optical metro networks and introduce the SDN concept in their deployment. CONTENT presents two general use cases: Infrastructure and network sharing (created logical resources) and cloud service provisioning on top of virtual infrastructures (end-to-end).

The integration of SDN with NFV is proposed in T-NOVA [36] and UNIFY [37] projects. On one hand, T-NOVA provides a framework to deploy VNFs over network infrastructures. The innovation of this project consists of their NFV Apps marketplace, which enables the easy creation, deployment and management of virtual network appliances in a standardized environment. T-NOVA proposes three general scenarios: High-Level Scenario, T-NOVA VNFs, and VNF Chaining. On the other hand, the UNIFY project takes advantage of cloud computing and the virtualization concept to provide a novel network architecture with optimized data traffic flows and the dynamic placement of networking, computer and storage components. This project presents eleven use cases, organized around the following domains: Infrastructure Virtualization, Flexible Service Chaining and Network Service Chain Invocation for Providers.

In the area of SDN and SON, CROWD [22] includes these technologies to enhance the coordination process between radio base stations in very dense and heterogeneous wireless networks (Dense Nets). This project allows the network cooperation, the dynamic network configuration, dynamic backhaul reconfiguration, energy optimization, etc. CROWD also presents fifteen use cases divided into two big scenarios: self-optimising dense networks and Optimised mobility in dense radio access networks. As part of the Horizon 2020 programme, 5G-NORMA [38] is a research project which aimed to provision an adaptive and open 5G infrastructure with capabilities to service customization, enhanced performance and security. To this purpose, this project introduces adaptability capacity to allocate mobile network functions in the most appropriate location and in a short time. Likewise, METIS II [39] presents a novel 5G RAN design, introducing a protocol stack architecture intended to provide a seamless integration of 5G radio technologies. The innovations of METIS II are focused on the spectrum management, air interfaces harmonization, resource management and a common control and user plane framework. The integration of them will support regulatory and standardization bodies. Other ongoing H2020 projects that combine SDN and NFV technologies are CHARISMA [40] and SONATA [41]. CHARISMA will enable the deployment of an intelligent cloud radio access network (C-RAN) and virtualized Customer Premise Equipment (CPE). SONATA will support network function chaining and an enhanced orchestration process in order to allow service customization.

The provision of innovative hardware and software platforms to support 5G infrastructures is proposed in FLEX5GWARE [42]. This project attempts to develop and prototype key components of 5G networks in the hardware and software domains. The main objective of this project is to deliver a highly reconfigurable hardware platform together with a well suited software platform, over which network elements and devices can be deployed following a modular, efficient and scalable approach. Several components must be deployed as 5G enablers, such as MIMO emulators, high-speed broadband converters, Filter Bank Multi-Carrier (FBMC) transceivers, Low-Density Parity Check (LDPC) codes, etc., with suitable interfaces to allow flexible software-based management schemes.

The integration of SDN, SON, NFV and Artificial Intelligence is encompassed by the SELFNET [43,44] project, which introduces intelligent, self-organizing and autonomic capacities to 5G networks, taking advantage not only of SDN and SON but also NFV and Cloud Computing. This project will provide a scalable, extensible and smart architecture to foster innovation and decrease

capital and operational expenditures derived from network management tasks. Moreover, SELFNET introduces the SON concept to facilitate the automatic management of network infrastructures. SON solutions are typically classified into three domains: self-protection, self-optimization and self-healing, which are the use cases proposed by SELFNET. Likewise, the COGNET Project [45] proposes the introduction of machine Learning, SDN and NFV in order to enhance monitoring tasks and autonomic network management. COGNET predicts the resource demand requirements and then changes its own configuration based on the network analysis (prediction, frauds, detecting error and security conditions).

Table 1 shows the current European use-case driven projects that tackle different 5G requirements, through a combination of SDN, NFV, SON and cloud computing concepts. All of these projects take into account SDN in different domains, such as e-health services, security, service chaining, multimedia optimization, etc.

Table 1. Research Projects in Mobile Networks.

Project Name	Related Technologies	Main Objective	Scenarios/Use Cases
MCN [34]	• SDN • Cloud Computing	Enhanced traffic processing by means of the separation between radio hardware and packet forwarding hardware.	• Cloud Computing for Mobile Network Operations • End-To-End Mobile Cloud
T-NOVA [36]	• SDN • NFV	Design and implementation of an integrated architecture for the automated provision and management of VNF infrastructures.	• High-Level Scenario • VNFs • Service chaining
UNIFY [37]	• SDN • NFV	The development of an automated and dynamic service provision platform, based on a service chaining architecture	• Infrastructure Virtualization • Flexible Service Chaining • Network Service Chain Invocation for Providers
CROWD [22]	• SDN • SON	The creation of technologies to support dynamic network functionality configuration and fine, on-demand, capacity tuning.	General scenario
5G-NORMA [38]	• SDN • NFV	The development of an adaptive, customizable, secure and efficient mobile network architecture to deal with complex traffic demand fluctuations.	• Multi-service • Multi-tenancy
CHARISMA [40]	• SDN • NFV	The creation of an intelligent and hierarchical routing and paravirtualized architecture to enhance end-to-end services.	General scenario
SELFNET [44]	• SDN • NFV • SON • Cloud	The design and implementation of an autonomic network management framework to achieve self-organizing capabilities in managing 5G network infrastructures, leveraging an improvement in the overall user experience.	• Self-healing • Self-protection • Self-optimization
COGNET [45]	• SDN • NFV • Machine Learning	Dynamic adaptation of the network resources (virtual network functions), while minimizing performance degradations and fulfill SLA requirements.	• Situational Context • Just-in-time Services • User-Centric Services • Optimized Services • SLA Enforcement • Collaborative Resource Manage

Last, but not least, it is worth mentioning the research efforts in the Americas, where a group of telecommunication service providers and manufacturers created the 5G Americas [46], an organization intended to foster the development of LTE wireless technology leveraging the adoption of 5G in the North and South America's society. At the same time, several activities have been conducted by academia. For instance, the Berkeley Wireless Research Center (BWRC) involves university, industry, government and other research stakeholders focused on exploring innovations in wireless communication systems based on radio frequency and millimeter wave technologies, which are its main challenge to develop reconfigurable radio architectures. Likewise, the Broadband Wireless Access and Applications Center (BWAC) involves around fifty research centers with the aim to collaborate with the industry in the creation of innovative and scalable wireless networks.

4. Future Trends and Challenges

The current necessities address the direction of the business and the requirements of Future Networks. It is expected that 5G networks will cover the increase of traffic volume by means of improving spectrum utilization, enhanced energy efficiency mechanisms, resource virtualization, resource sharing, self-management and self-organization capabilities [8]. The concept of Future Networks envisages a broad range of opportunities in different fields. In other words, it will cover not only the traditional network fields but also other domains, such as e-health, energy efficiency, emergency services, public safety, IoT, machine-to-machine (M2M) communication, Information Centric Networking (ICN), among others. The applicability of SDN, NFV, SON and cloud computing opens the door to facilitate the deployment and management of services in an open business environment. Indeed, if we take into account technologies and advances of different initiatives, a possible architecture for future networks is proposed (Figure 4). On one hand, it presents a layered structure: infrastructure, virtualization, control and application layers, similar to the SDN approach. On the other hand, VNFs and NFV M&O modules are incorporated in order to control the NFVI. For its part, cloud technologies are present on the cloud computing layer and SON capacities will aid in the decision process in the control layer.

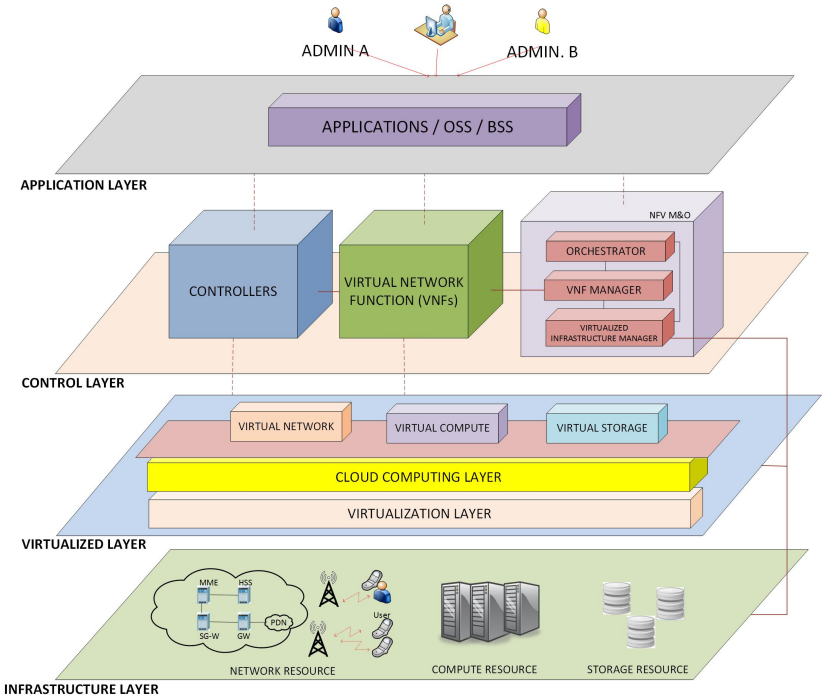

Figure 4. Future Network Architecture.

Future Networks could incorporate all of these concepts or part of them. Despite of the advantages of this proposal, there are some challenges that need to be overcome in order to successfully combine these technologies. Firstly, the unified definition and standardization in the separation of the data plane and control plane and the provision of virtualized instances will enable the easy development and integration of the future network technologies. In addition, the complexity of the mobile network elements constitutes a big challenge by itself. At the same time, these kind of systems will require effective pricing schemes and business models with two objectives in mind: (i) Customers pay only for the provisioned service and (ii) stakeholders receive revenues according to their SLAs. Another important issue is how legacy networks will coexist with new systems, which is still a relatively unexplored field.

In the management and orchestration field, significant changes are required not only to improve the processing of data information but also to optimize the deployment and allocation of network resources. A unified management framework could allow enhanced traffic monitoring, provide self-management capabilities and network customization. A virtualized environment faces some issues, such as finding the best place to allocate virtual functions (operator infrastructure or cloud), migration and scheduling process. Mechanisms are also needed to provide load balancing, energy efficiency algorithms, inter-domain capabilities, among others. In parallel, all of these characteristics should be provided in a secure and trusted environment with enhanced capacities to recovery from failures. Moreover, The SDN centralized control or the dynamism of cloud computing are challenges that need to be covered. Table 2 shows the challenges and future trends that must be covered in order to fulfill the user needs of 5G networks.

Table 2. Current Trends and Challenges.

Requirement	Challenge	Future Trends/Enabler Technologies
System Performance	• Provide efficient mechanisms regarding to radio resource provisioning. • Improving the capacity of radio resources. • Provide super wide bandwidth. • Better management of data traffic, interference and mobility levels.	• Evolution of radio-access technologies (RATs). • Decreasing the cell size. • Millimeter-wave communication • Intelligent resource allocation via SDN or SON.
Composite Wireless Infrastructures	The 5G device can choose the most appropriate wireless or mobile technology according their needs (Change between systems).	• Enhancement of user devices (Muti-Band-Multi-Mode support). • Introduction of intelligent mechanisms and SDN control.
Facilitating very dense deployments (Hetnets)	Operators must provide effective mechanisms to deploy cells of different sizes according to user needs.	• Improving the resource capacity through decreasing the cell size. • Introduction of intelligent and Software Defined Radio (SDR) concepts.
Flexible spectrum management	Improve the spectrum utilization in order to operate in some spectrum bands or channels, while reducing interferences.	• Massive MIMO • Mechanisms to use unused bands.
Native support D2D Communication	Deploy networks based on interconnected end user devices (machines, sensors, etc). The traffic will be properly assigned without cause congestions.	• Introduction of Cognitive Intelligent mechanisms to exchange traffic between users. • Smarter end-user devices.
Reduce Capex and Opex	• Reduce the average service creation. • Dynamic scalability and deployment of services and NFs, while reducing the complexity in planning and configuration tasks.	• Resource sharing (Exploring Cloud-RAN, Cloud computing, NFV) • Smarter allocation of functional mobile components (SDN, NFV).

Table 2. *Cont.*

Requirement	Challenge	Future Trends/Enabler Technologies
Muti-tenancy and multi-service support	Service providers can control the resources deployed in a shared infrastructure (network, computing, mobile resources).	• Cloud computing • SDN • NFV • Mobile Edge Computing (MEC)
Open Environment	New applications and NFs could be deployed in an open environment, regardless of the network hardware and technologies used by operators.	• Standardization of SDN and NFV concepts. • Introduction of SDR.
Energy efficiency operation	Saving energy per service provided. Nowadays, most of the energy consumption comes from RAN elements.	Introduction of intelligent and SON capabilities taking into account the device status.
Monitoring and Management	Provide self-management and self-optimization capabilities to 5G systems.	• Automated management and monitoring functions (SDN, NFV). • Takes decisions based on historical record of network status.
Ensuring QoS/ QoE and SLA	A 5G user will be able to obtain enhanced services, regardless of the location or network technologies (compared with 4G systems), for several use cases such as emergency situations or network failures.	• Enhanced mechanism to monitor the network status (traffic optimization techniques) via SDN and intelligent mechanism. • Automated network configuration to ensure the required need (SDN, NFV).
Charging and billing	Create different user profiles so that customers pay only the required service (pay-as-you-go), while operators bill the respective service.	Introduction of SDN and NFV concepts.

It is important to note the current efforts of initiatives such as 5G Americas [46], 5G-PPP or NGMN to develop future network. They promote not only SDN, NFV and cloud computing adoption but also the study of transversal concepts such as carrier aggregation, massive MIMO, Multi-RAT convergence, spectral and signaling efficiency, among others. It is imperative that telecommunication and network service providers find a consensus to develop solutions, architectures, technologies and standards for the next generation of infrastructures. The communication paradigm of anytime, anyhow and anywhere will become a reality in the future society.

5. Conclusions

The next generation of mobile networks will be able to support higher capacity, lower latency and massive network access compared with current mobile deployments. Future mobile systems require the enhancement of radio and network elements, which takes advantage of intelligent mechanisms, cloud computing, SDN and NFV approaches. At the same time, these technologies foster the service innovation and provide ubiquitous and on-demand features. This paper has discussed SDN, NFV, Cloud Computing and SON concepts as enabler technologies to design future mobile networks. We have focused on concepts that could aid to cover operational and management requirements as well as radio and network capabilities. It also presents the ongoing projects and use cases that leverage these technologies in the context of mobile networks. This work is intended to explain the benefits that we could obtain from the combination between these concepts and the challenges to lay the foundations of Future Mobile Networks.

Acknowledgments: This work is supported by the European Commission Horizon 2020 Programme under grant agreement number H2020-ICT-2014-2/671672 - SELFNET (Framework for Self-Organized Network Management

in Virtualized and Software Defined Networks). Lorena Isabel Barona López and Ángel Leonardo Valdivieso Caraguay are supported by the Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación SENESCYT (Quito, Ecuador) under Convocatoria Abierta 2012 and 2013 Scholarship Program.

Author Contributions: The authors contributed equally to this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. European Commission. Future Internet 2020: Visions of an Industry Expert Group 2009. Available online: http://www.future-internet.eu/fileadmin/documents/reports/FI_Panel_Report_v3.1_Final.pdf (accessed on 16 December 2016).
2. Nokia Network. 5G Use Cases & Requirements. 2014. Available online: http://networks.nokia.com/file/31121/5g-requirements (accessed on 16 December 2016).
3. Agyapong, P.K.; Iwamura, M.; Staehle, D.; Kiess, W.; Benjebbour, A. Design Considerations for a 5G Network Architecture. *IEEE Commun. Mag.* **2014**, *52*, 65–75.
4. NGMN Alliance. NMGN 5G White Paper 2015. Available online: https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf (accessed on 16 December 2016).
5. Boccardi, F.; Heath, R.W.; Lozano, A.; Marzetta, T.L.; Popovski, P. Five Disruptive Technology directions for 5G. *IEEE Commun. Mag.* **2014**, *52*, 74–80.
6. Gupta, A.; Jha, R.K. A Survey of 5G Network: Architecture and Emerging Technologies. *IEEE Access* **2015**, *3*, 1206–1232.
7. Bangerter, B.; Talwar, S.; Arefi, R.; Stewart, K. Networks and Devices for the 5G Era. *IEEE Commun. Mag.* **2014**, *52*, 90–96.
8. Andrews, J.G.; Buzzi, S.; Choi, W.; Hanly, S.V.; Lozano, A.; Soong, A.C.; Zhang, J.C. What Will 5G Be? *IEEE J. Sel. Areas Commun.* **2014**, *32*, 1065–1082.
9. Demestichas, P.; Georgakopoulos, A.; Karvounas, D.; Tsagkaris, K.; Stavroulaki, V.; Lu, J.; Xiong, C.; Yao, J. 5G on the Horizon: Key Challenges for the Radio-Access Network. *IEEE Veh. Technol. Mag.* **2013**, *8*, 47–53.
10. Kreutz, D.; Ramos, F.M.V.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-defined Networking: A Comprehensive Survey. *Proc. IEEE* **2015**, *103*, 14–76.
11. ETSI Industry Specification Group (ISG). Network Function Virtualization (NFV) Architectural Framework. 2013. Available online: http://www.etsi.org/technologies-clusters/technologies/nfv (accessed on 16 December 2016).
12. Zhang, Q.; Cheng, L.; Boutaba, R. Cloud Computing: State-of-the-art and Research Challenges. *J. Int. Ser. Appl.* **2010**, *1*, 7–18.
13. Baldo, N.; Giupponi, L.; Mangues-Bafalluy, J. Big Data Empowered Self Organized Networks. In Proceedings of the 20th European Wireless Conference, Barcelona, Spain, 14–16 May 2014.
14. Buda, T.S.; Assem, H.; Xu, L.; Raz, D.; Margolin, U.; Rosensweig, E.; Lopez, D.R.; Corici, M.I.; Smirnov, M.; Mullins, R.; et al. Can Machine Learning aid in delivering new Use Cases and Scenarios in 5G? *IEEE Netw. Oper. Manag. Symp. (NOMS)* **2016**, 1279–1284, doi:10.1109/NOMS.2016.7503003.
15. Mijumbi, R.; Serrat, J.; Gorricho, J.L.; Bouten, N.; Turck, F.; Boutaba, R. Network Function Virtualization: State-of-the-art and Research Challenges. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 236–262.
16. Contreras, L.M.; Doolan, P.; Lonsethagen, H.; López, R. Operational, Organizational and Business Challenges for Network Operators in the context of SDN and NFV. *Comput. Netw.* **2015**, *92*, 211–217.
17. Hu, F.; Hao, Q.; Bao, K. A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1–27.
18. McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J. OpenFlow: Enabling Innovation in Campus Networks. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 69–74.
19. Sefraoui, O.; Aissaoui, M.; Eleuldj, M. OpenStack: Toward an Open-source Solution for Cloud Computing. *Int. J. Comput. Appl.* **2012**, *55*, 38–42.
20. Sonkoly, B.; Németh, F.; Csikor, L.; Gulyás, L.; Gulyás, A. SDN Based Testbeds for Evaluating and Promoting Multipath TCP. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, Australia, 10–14 June 2014; pp. 3044–3050.

21. Pentikousis, K.; Wang, Y.; Hu, W. MobileFlow: Toward Software-defined Mobile Networks. *IEEE Commun. Mag.* **2013**, *51*, 44–53.
22. EU CROWD Project. Connectivity Management for EneRgy Optimised Wireless Dense Networks. Project Reference: 318115. Funded under: FP7-ICT. Available online: http://www.ict-crowd.eu/ (accessed on 16 December 2016).
23. Abdelwahab, S.; Hamdaoui, B.; Guizani, M.; Znati, T. Network Function Virtualization in 5G. *IEEE Commun. Mag.* **2016**, *54*, 84–91.
24. Open Networking Foundation (ONF), 2016. Available online: https://www.opennetworking.org/ (accessed on 16 December 2016).
25. ETSI Industry Specification Group (ISG). Network Function Virtualization Use Cases 2013. Available online: http://www.etsi.org/deliver/etsi_gs/nfv/001_099/001/01.01.01_60/gs_nfv001v010101p.pdf (accessed on 16 December 2016).
26. Open Platform for NFV (OPNFV). Available online: https://www.opnfv.org/ (accessed on 16 December 2016).
27. Floodlight SDN Controller. Available online: https://github.com/floodlight/floodlight (accessed on 16 December 2016).
28. OpenDaylight SDN Controller. Available online: https://www.opendaylight.org/start (accessed on 16 December 2016).
29. 5G Infrastructure Public Private Partnership - 5G PPP. Available online: https://5g-ppp.eu (accessed on 16 December 2016).
30. IMT-2020 (5G) Promotion Group. Available online: http://www.imt-2020.cn/en/introduction (accessed on 16 December 2016).
31. 5G Forum. Available online: http://www.5gforum.org/ (accessed on 16 December 2016).
32. Fifth Generation Mobile Communication Promotion Forum (5GMF). Available online: http://5gmf.jp/en/ (accessed on 16 December 2016).
33. Osseiran, A.; Boccardi, F.; Braun, V.; Kusume, K.; Marsch, P.; Maternia, M.; Queseth, O.; Schellmann, M.; Schotten, M.; Taoka, H.; et al. Scenarios for 5G Mobile and Wireless Communications: The Vision of the METIS Project. *IEEE Commun. Mag.* **2014**, *52*, 26–35.
34. MCN Project. Funded under: FP7-ICT. Project Reference: 318109, Funded under: FP7-ICT. Available online: http://www.mobile-cloud-networking.eu/site/ (accessed on 16 December 2016).
35. CONTENT Project. Convergence of Wireless Optical Network and iT rEsources iN SupporT of Cloud Services. FP7-ICT. Project Reference: 318514, Funded under: FP7-ICT. Available online: http://cordis.europa.eu/fp7/ict/future-networks/documents/call8-projects/content-factsheet.pdf (accessed on 16 December 2016).
36. EU T-NOVA Project. Network Functions as-a-Service over Virtualised Infrastructures. Project Reference: 619520. Funded under: FP7-ICT. Available online: http://www.t-nova.eu/ (accessed on 16 December 2016).
37. EU UNIFY Project. Unifying Cloud and Carrier Networks. Project Reference: 619609. Funded under: FP7-ICT. Available online: http://www.fp7-unify.eu/ (accessed on 16 December 2016).
38. 5G-NORMA Project. 5G NOvel Radio Multiservice Adaptive Network Architecture. Project Reference: 671584. Funded under: H2020-ICT-2014-2. Available online: https://5gnorma.5g-ppp.eu/ (accessed on 16 December 2016).
39. METIS-II Project. Mobile and Wireless Communications Enablers for Twenty-Twenty (2020) Information Society-II. Project Reference: 671680. Funded under: H2020-ICT-2014-2. Available online: https://5g-ppp.eu/metis-ii/ (accessed on 16 December 2016).
40. CHARISMA Project. Converged Heterogeneous Advanced 5G Cloud-RAN Architecture for Intelligent and Secure Media Access. Project Reference: 671704. Funded under: H2020-ICT-2014-2. Available online: http://www.charisma5g.eu// (accessed on 16 December 2016).
41. SONATA Project. Service Programing and Orchestration for Virtualized Software Networks. Project Reference: 671517. Funded under: H2020-ICT-2014-2. Available online: http://www.sonata-nfv.eu// (accessed on 16 December 2016).
42. Flex5Gware Project. Flexible and Efficient Hardware/Software Platforms for 5G Network Elements and Devices. Project Reference: 671563. Funded under: H2020-ICT-2014-2. Available online: http://www.flex5gware.eu/ (accessed on 16 December 2016).

43. Neves, P.; Calé, R.; Costa, M.R.; Parada, C.; Parreira, B.; Alcaraz-Calero, J.; Wang, Q.; Nightingale, J.; Chirivella-Perez, E.; Jiang, W.; et al. The SELFNET Approach for Autonomic Management in an NFV/SDN Networking Paradigm. *Int. J. Distrib. Sens. Net.* **2016**, *2016*, 1–17.
44. SELFNET Project. Framework for Self-Organized Network Management in Virtualized and Software Defined Networks. Project reference: 671672. Funded under: H2020-ICT-2014-2. Available online: https://selfnet-5g.eu/ (accessed on 16 December 2016).
45. Xu, L.; Assem, H.; Yahia, I.G.B.; Buda, T.S.; Martin, A.; Gallico, D.; Biancani, M.; Pastor, A.; Aranda, P.A.; Smirnov, M.; et al. CogNet: A Network Management Architecture Featuring Cognitive Capabilities. *IEEE Netw. Commun. (EuCNC)* **2016**, *2016*, 325–329.
46. 5G Americas, 2016. Available online: http://www.5gamericas.org/es/ (accessed on 16 December 2016).

future internet

MDPI

Article

Towards Incidence Management in 5G Based on Situational Awareness

Lorena Isabel Barona López †, Ángel Leonardo Valdivieso Caraguay †, Jorge Maestre Vidal †, Marco Antonio Sotelo Monge † and Luis Javier García Villalba *,†

Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), Faculty of Computer Science and Engineering, Office 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, Spain; lorebaro@ucm.es (L.I.B.L.); angevald@ucm.es (Á.L.V.C.); jmaestre@ucm.es (J.M.V.); masotelo@ucm.es (M.A.S.M.)

* Correspondence: javiergv@fdi.ucm.es; Tel.: +34-91-394-7638

† These authors contributed equally to this work.

Academic Editor: Dino Giuli

Received: 11 September 2016; Accepted: 5 January 2017 ; Published: 17 January 2017

Abstract: The fifth generation mobile network, or 5G, moves towards bringing solutions to deploying faster networks, with hundreds of thousands of simultaneous connections and massive data transfer. For this purpose, several emerging technologies are implemented, resulting in virtualization and self-organization of most of their components, which raises important challenges related to safety. In order to contribute to their resolution, this paper proposes a novel architecture for incident management on 5G. The approach combines the conventional risk management schemes with the Endsley Situational Awareness model, thus improving effectiveness in different aspects, among them the ability to adapt to complex and dynamical monitoring environments, and countermeasure tracking or the role of context when decision-making. The proposal takes into account all layers for information processing in 5G mobile networks, ranging from infrastructure to the actuators responsible for deploying corrective measures.

Keywords: 5G; incidence management; information security; SDN/NFV; Situational Awareness

1. Introduction

The rapid proliferation of the use of mobile devices has revealed the lack of ability of the current networks to accommodate the vast amount of information that they will have to manage [1–3]. This situation has given rise to the development of a brand new generation of mobile networks not only to provide solutions to such problems, but also to improve many features of their predecessors. Enhanced capabilities, related to transfer massive data, interoperability or reduction in energy consumption, allow a better Quality of Experience (QoE) to the users [4]. Achieving these goals requires great capacity for innovation, such as high speed data transfers or better information management methods [5]. The last part has significant impact on business models based on services and real-time applications (e-health, e-security, Voice over IP, streaming, etc.), where emerging technologies, such as Software-Defined Networking (SDN) or Network Function Virtualization (NFV), facilitate pattern customization and management of the mobile network traffic. However, the development of these services is limited by the poor performance in the management and decision-making strategies [6], which additionally entails difficulty when deploying information security measures, the principal objective of this research. Nowadays, information security management plays a major role towards achieving the objectives and goals of companies and organizations. Traditionally, it has been carried out by implementing guidelines, standards and platforms that aim to protect their resources and assets (ISO/IEC 27000 [7], NIST-SP 800 [8], CVSS-SIG-First [9], MAGERIT [10], ITIL and COBIT [11], etc.).

However, these proposals have shown shortcomings when they are implemented in dynamic scenarios, where the context plays a very important role in decision-making [12]. This is the case of network-based monitoring environments, and more specifically, those that implement 5G technology, where the assets and events are considered highly dependent on the environment. As a solution to this problem, some authors have adopted incident management methodologies capable of handling information in a much more cognitive way, and, therefore, facilitating their understanding through contextual analysis. Worthy of special mention are those based on constructing Situational Awareness (SA) of the protected environment by applying the Endsley's model, where the perception, comprehension and projection of the system status are kept in mind [13]. The adaptation of this paradigm to the management of information security in networks has led to the coining of the term Network Security Situational Awareness (NSSA) [14]. Despite, however, its effectiveness having been proven in existing networks, it has not yet been considered to meet the challenges posed by 5G technologies.

To the best of our knowledge, there are few studies that survey Security and Risk Management in 5G Networks. In [15], a context aware framework for the next generation of Mobile Cloud Network (MCN) is proposed. This work introduces a "Context Generation and Handling Function" to provide enriched processing information from radio and core elements, taking into account two key-enabled 5G technologies (SDN and NFV concepts). Meanwhile, a recently published threat report has been conducted by the European Union Agency for Network and Information Security (ENISA) [16]. This work reviews the potential security in SDN/5G networks, considering not only SDN but also NFV and Radio fields. This report identifies the network assets and the security threats, their related challenges and risks. It also describes the existing security methods and provides good practices for 5G systems. These works could be considered part of the initial research of 5G Security Management. However, these are limited in scope. On one hand, the ENISA Report [16] identifies only the assets and threats to SDN/5G environments (no architecture proposal has been done). On the other hand, Marquezan et al. [15] propose a single network function that monitors radio and access elements but doesn't take into account other 5G components such as virtualization or application layers. Meanwhile, the 5G-Ensure Project is intended to cover security requirements in 5G Networks. The proposed architecture will provide a trustworthy 5G system, offering reliable security services to customers by means of the development of a set of non-intrusive security enablers such as privacy, security network management, and trust, among others [17]. As part of its proposal, 5G-Ensure defines a Risk Assessment and Mitigation methodology in order to evaluate security concerns in 5G systems, based on NIST-SP-800-30 and ISO 27005 standards [18]. Although 5G-Ensure covers a wide range of security issues on 5G Networks, this project is still at an early stage and does not keep in mind the concept of actuators, which have been introduced in our proposal in order to mitigate possible risks and deploy corrective measures. This article introduces a novel architecture for incident management on 5G Mobile networks, which combines the foundations of the traditional risk management guidelines with the Situational Awareness model published by Endsley. It covers all layers of information processing in 5G networks, from the infrastructure to the actuators responsible for implementing mitigation actions. The basis for the identification, monitoring, analysis, decision-making, prediction and countermeasure tracking are also introduced. The paper is divided into six sections, the first section being this introduction. The risk management systems, Situational Awareness and the related work are described in Section 2. The challenges posed by 5G networking related to risk management are explained in Section 3. An architecture that combines traditional risk management and Situational Awareness for 5G is detailed in Section 4. The componets to carry out analysis and decision-making processes are described in Section 5. Finally, conclusions and future work are presented in Section 6.

2. Background

The following describes the main characteristics of Information Security Risk Management (ISRM), Situational Awareness and the related works.

2.1. Information Security Incidence Management

The problems inherent to incidence assessment and management have captured the attention of the research community over the recent four decades. Consequently, several studies have been published in order to collect the most relevant contributions, as well as to identify the causes that have given way to their evolution, as it is described in [19,20]. The bibliography covers a very large collection of topics that range from the definition of risk and its scientific approach [21] to discussions about their development by different governments and organizations [22]. On the other hand, the need to protect the information technology has also led to the publication of standards [7–9] and guidelines [9,11] for their proper implementation. Most of these proposals agree that the incident management process must be carried out in the following steps: framing, assessing, monitoring and responding [20]. Risk framing determines the context and a common perspective on how organizations manage risk, which include their goals, policies, constraints, risk tolerance, priorities, trade-offs, and principles of action [7]. Incidence assessment identifies potential risks, organizational assets and vulnerabilities. Then, it evaluates the risks according to different criteria, such as likelihood of occurrence, capacity to inflict harm or dimensions (confidentiality, integrity, availability, authenticity, etc.) [9,10]. Because of its complexity [23], this is the step with the largest presence in the literature, which has motivated the development of specific systems for Information Security Risk Assessment (ISRA). In the monitoring step, the ISRM maintains awareness of the incidences being incurred, which implies looking for events in the monitoring environment that allow the detection of the previously determined threats. If any of them is identified, the response tasks decide and apply the appropriate countermeasures.

Due to the large differences between monitoring scenarios, the success of decision-making depends directly on the previously described processes and their ability to adapt to each use case. The following are examples of methodologies to facilitate their integration into more specific monitoring environments: Cloud Environments [24], industrial control [25], embedded systems [26] and Supervisory Control And Data Acquisition (SCADA) [27]. Given the nature of 5G devices and layers, it is important to track suspicious events in order to provide more intelligence to 5G security and monitoring systems [28,29]. Forensics are intended to collect, analyze and interpret digital data connected to a security incident in order to track an evidence trail (how an attack was carried out or how an event occurred). In the context of 5G mobile networks, these incidents might be related to physical and virtual devices as well as sensors, SDN/NFV functions and cloud elements [30].

2.2. Situational Awareness and Information Security

As defined by Endsley, the term Situational Awareness (SA) refers to "the perception of the elements in the environment within a volume of time and space, comprehension of their meaning and the projection of their status in the near future" [31]. Usually, this definition is simplified as "knowing what's going on so you can figure what to do" [32]. Thus, it is clear that its aim is to facilitate decision-making based on what is happening and its projection [13]. In order to acquire Situational Awareness, Endsley proposes three stages of information processing: perception, comprehension and projection; the first conducts the tasks of monitoring and identification of incidences, the second their analysis and association, and the last predicts the evolution of the state of the system. As shown in Figure 1, once relevant situations are detected, the countermeasures to be applied are decided and executed. It is important to highlight that there is feedback between action/decision levels and the Situational Awareness; in this way, the countermeasures and their impact on the system are tracked. The observed results have implications on future decisions and facilitate the use of advanced diagnostic methods [14].

Incident management based on Situational Awareness has been implemented in very different areas, among them smart grids [33], power generation [34] or vehicular collision avoidance systems [35]. In [36], a method for defining the critical information and the relevant information quality elements that are required to build the Shared Situational Awareness (SSA) in disaster response is suggested. The adaptation of the Endsley has proved particularly effective in complex

and dynamic environments [37], where the diagnosis is highly dependent on the context in which incidents are reported, reaching to play an essential role in the fight against cybercrime. Many of these contributions are collected in [38], where the predominance of issues related to risk management in emergency situations, industrial systems and networks is observed. As discussed in [12], they improve the three most repeated deficiencies of the Information Security Risk Management: (1) Information security risk identification is commonly perfunctory; (2) Information security risks are commonly estimated with little reference to the current situation; and (3) Information security risk assessment is commonly performed on an intermittent, non-historical basis (a conventional security risk assessment scheme can only give a "snapshot" of the risks of the information systems at a particular time [39]). In order to bring solutions to these problems, but without losing focus on the ISRM/ISRA basis, several publications approach the combination of both paradigms. This is the case of [12,40], where the Situational Awareness is acquired, taking into account the definition of risks, assets and their impact posed by the different standards and platforms for ISRA implementation [7–11].

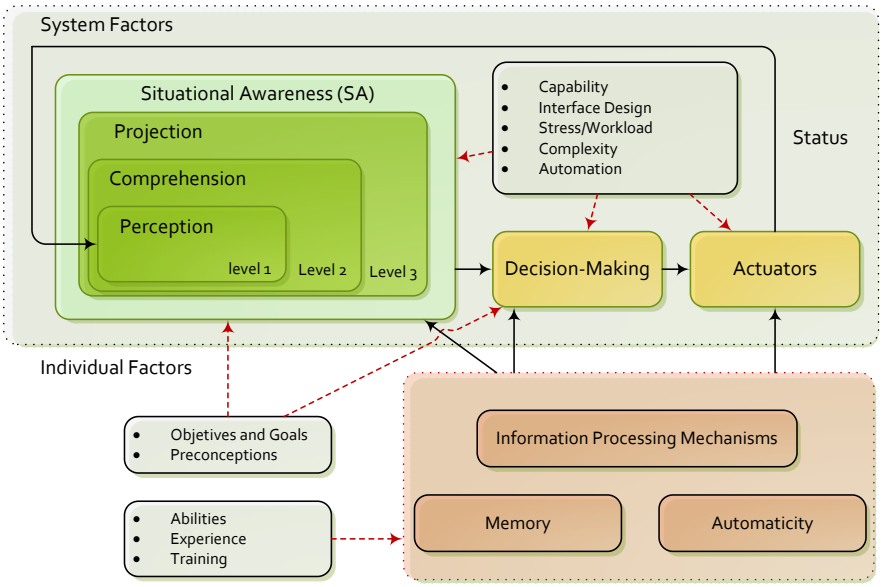

Figure 1. Endsley model for Situational Awareness.

3. Incidence Management in 5G

The new 5G design principles are intended to support an exponential increase of connected devices and, consequently, the data traffic moving through the network. In contrast to traditional mobile architectures, 5G requests a clear separation between data and control planes, a global vision of the network and a dynamic/customizable control of the mobile network operations. For this purpose, innovative technologies, such as SDN and NFV, have been extended to wireless and mobile platforms. In this way, the operators are not limited by the use of Command Line Interface (CLI) for individual and remote access. Instead, the administrator can create software or "network applications" to dynamically control the network behavior. However, the use of autonomous incident management systems that take advantage of these new paradigms is limited. In this context, the main challenge is the coordination between the virtual monitoring elements, allocated in different nodes in the infrastructure, and the response or mitigation procedures through the execution of actions in virtual functions. Similarly, mobile SDN/NFV-enabled architectures are limited by the lack of integrated schemes capable of analyzing large volumes of data, detecting potential risks and diagnosing their causes. Furthermore, the management systems should enable the definition, organization and handling of the different risks, assets and priorities without compromise the security and quality of service.

4. Information Security Architecture for 5G

The proposed architecture is mainly focused on autonomous risk management of 5G mobile infrastructures based on SDN/NFV architectures. In order to establish the coverage and limits of this approach, the following addresses the assumptions and requirements of the following design:

- The elements responsible for monitoring and executing mitigation actions (e.g., virtual functions) are compatible with the SDN/NFV paradigm. If the elements follow the traditional architectures, a compatibility layer is assumed. This additional layer can use the available configuration options to emulate a SDN/NFV enabled element.
- The communication between the different modules of the framework must be performed through secure channels.
- The information provided by the monitoring elements (low level metrics, alerts) are considered reliable.
- The Risk Analysis and the corresponding Situation Awareness procedures are strongly isolated from the data plane forwarding. In other words, the resources (network, storage and computing) used for the operation of the framework belong to administrative domain and, consequently, do not modify the capabilities of the 5G forwarding elements.
- The functional modules are extensible and can be implemented using distributed architectures in function of the available management resources and the size of the managed infrastructure.

The proposed architecture presents the synergy between 5G risk analysis and the Endsley model, and is depicted in Figure 2. The model describes four functional layers: Virtual Infrastructure and Sensors, Monitoring and Correlation, Analysis, and Decision-Making and Actuators. The Virtual Infrastructure executes the data forwarding engine and Sensors monitors the different metrics of the network. For its part, the perception, comprehension and projection principles of the Situational Awareness are applied in the Monitoring, Correlation and Analysis modules. The Decision-Making and Actuators complete the circle with the execution of proactive and reactive actions to optimize and solve problems located in the virtual infrastructure. The following sections of this chapter focus on the description of the Virtual Infrastructure/Sensors and Monitoring/Correlation.

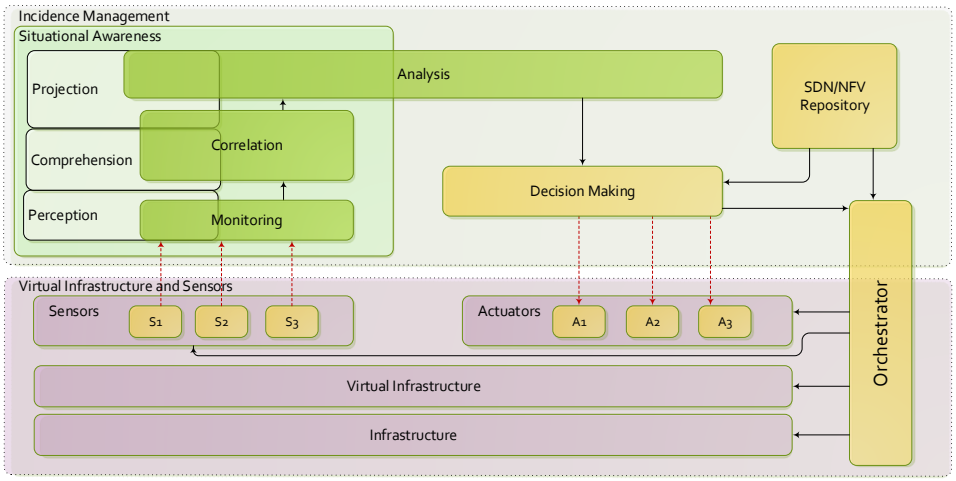

Figure 2. Endsley model for Situational Awareness.

4.1. Virtual Infrastructure and Sensors

The main purpose of this layer is the abstraction of the different hardware/software elements running in the mobile infrastructure and enabling the monitoring of low-level metrics related to the network behavior/status. Its developments includes the innovative designs principles of 5G networks: decoupling

of data and control planes, virtualization of mobile functions, and a complete integration with cloud computing environments. In this way, the SDN architecture promotes the separation between data and control planes in mobile infrastructure (base stations, links, servers, gateways, Deep Packet Inspection (DPI), among others). The network administrator is not limited by the traditional private/closed hardware/software system and, consequently, the data forwarding engine can be customized. Similarly, the virtualization layer enables the dynamic allocation of virtual resources based on the user requirements. For its part, the NFV approach proposes the implementation of the different services (e.g., firewall, DPI, Quality of Service optimizer, load balancer) as virtual software functions that can be instantiated in different points of the virtualized infrastructure. As a result, the architecture considers two software applications types: SDN-Apps and NFV-Apps. SDN-Apps is executed in software programs to control the data plane in network devices and the NFV-Apps are virtual functions that can be instantiated in virtualized elements to develop a particular service. In the proposed architecture, the sensors are specialized NFV-Apps capable of monitoring different metrics on the system. Traffic analyzers, QoS analyzers, and anomalies/botnet/Distributed Denial of Service (DDoS) attack detectors are examples of sensors. These sensors (NFV-Apps) can be instantiated in different locations of the virtual infrastructure and reconfigured depending on the requirements of upper layers. Consequently, the system is able to increase the surveillance in suspected hazardous areas and establish quarantine regions.

4.2. Monitoring and Correlation

The monitoring module collects the information provided by lower layers (Virtualized Infrastructure and Sensors) and applies aggregation/correlation techniques to simplify the further analysis tasks:

- *Monitoring (Data Collection)*. The main objectives are the gathering and management of the information from all data sources, and facilitating their access to upper layers. Monitoring tasks would be able to actively poll different sources to collect real-time statistics, providing highly accurate and low overhead traffic measurement [41]. This module also controls the registration and access process of new sensors. The collected information is organized in efficient data structures, taking into account the large amount of data to be processed. In this regard, two scenarios were considered. In the first scenario, the sensor sends a report to the monitor when it detects relevant information (alerts, link failures, memory or CPU overload). In the second scenario, the monitor requests information (whenever necessary) to the sensors in order to facilitate the aggregation and analysis tasks (virtual topology, available links, among others).
- *Correlation*. It is responsible for the first abstraction level of information processing, in which, in order to have a global view of the network status, correlation and aggregation processes are executed. Information considered as redundant or non-sensitive is discarded. As an example, in case of multiple alerts received from each device belonging to the same affected area, a single alert is displayed with the affected topology. Due to the dynamism offered by virtual environments, in contrast to the rigidity of the physical elements, network topology is expressed as an extended or increased graph ($G_a(V_a, E_a)$), which models virtual nodes (V_a) and links (E_a) located in the physical infrastructure [42,43]. Likewise, as a result of correlation and aggregation operations, the received low-level metrics can be expressed or translated into high-level metrics, also known as Health of Network (HoN). For example, transmission data rate (Mbps), delay (ms) and jitter (ms) of data in streaming video, collected by the sensors at different points in the network, can be expressed as an overall perception of quality of service QoS/QoE, quantified by the measurement of the Mean Opinion Score (MOS).

5. Analysis and Decision-Making

This section describes the principal characteristics of the components related to analyzing the gathered information, decision of countermeasures and their deployment.

5.1. Analysis

The analysis component performs identification of network situations from metrics provided by the aggregation module and reaches diagnoses that contribute in decision-making tasks. In general terms, the analysis studies any aspect related with the incidences reported by the 5G use cases and the risks that could compromise the system requirements. In this context, situations are divided into two main groups: events and risks. Events are defined as situations that occur within 5G mobile networks which a priori do not display harmful features but are useful in diagnosis. The events are grouped into four categories: discovery, removal, modification and notification, which are described as follows:

- *Discovery events*—include all situations related to incorporation of new assets to the system. For example, this occurs when incorporating new nodes into the network, establishment of new connections between previously existing resources, or deployment of new virtualization layers. Each time a discovery event is communicated, the asset inventory is updated.
- *Removal events*. Unlike discovery events, removal events indicate situations related to the elimination of 5G resources. These are the cases of deletion of assets, removal of connections between nodes, or elimination of virtualization resources. As in discovery events, each time a removal event is communicated, the asset inventory is upgraded.
- *Modification events*. They include every situation related with the modification (not removal) of an existing resource. For example, this occurs when varying the location of the asset (i.e., changes to IP address, MAC address, etc.), and changes between communication protocols or software updates. As in the previous cases, modification events involve changes to the asset inventory.
- *Notification events*. They report specific situations in the network that are not related to changes in the assets inventory, such as periodic reviews of the bandwidth status, presence of unused resources or requests for special configurations.

On the other hand, risks are inherently damaging, and they may be inferred from network mapping or directly reported by the use case agents. An example of the first case is the identification of bottlenecks, congested regions or resources depletion. In the other case, a striking example relates to defensive use cases, where security NFV-Apps (Intrusion Detection Systems, honeypots, etc.) directly reports intrusions such as malware spreading or denial of service threats. The bases of 5G situation analysis are shown in Figure 3. They include: detection, risk assessment, asset inventory management, risk map, prediction, diagnosis and countermeasure tracking. The following briefly describes the most important features of each of them.

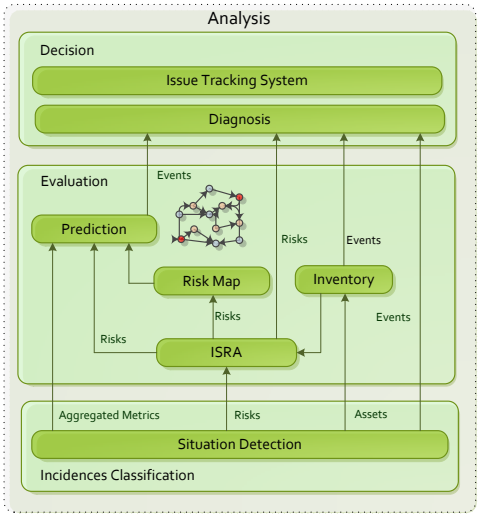

Figure 3. Situation analysis on 5G.

5.1.1. Detection

The detection module is the connection between monitoring/aggregation tasks and the functions for understanding the information. Its inputs are aggregated high-level metrics built from correlated data and reports of situations directly issued by the sensors. After processing this information, the detection module builds the primitive situations (events or risks) to be analyzed. The rules to infer situations from the perceived data are provided by security operators and determined after the risks/events identification. Note that the same combination of metrics could trigger different situations; because of this, the proper management of the detection component implies adaptation of expert systems, especially those based on rules.

5.1.2. Risk Assessment

Risk assessment combines some of the most widely spread strategies approved by the research community for Information Security Risk Assessment (ISRA), among them the guidelines ISO/IEC 27000 [7], NIST-SP 800 [8], CVSS-SIG-First [9] or MAGERIT [10]. The approach to identify risk factors assumes the basis on ISO/IEC 2700 series and NIST-SP800. However, since these are mainly aimed at general purpose risk assessment, they lack specificity; in particular, they do not take into account the 5G design principles, infrastructure or requirements. Because of this, they must be adapted to the 5G mobile networks circumstances. Another obstacle is that they are based on metrics that are too simple. In order to improve the ability to understand the impact and facilitate decision-making, as well as consider a more current model, a group of advanced measures similar to those proposed in CVSS-SIG-First should be adapted. Thereby, a larger amount of characteristics on the potential vulnerabilities should be studied, thus assuming the union of three metric sets that contain intrinsic (base), temporal and specific (environmental) features. On the other hand, approaches like MAGERIT provide alternative ways to calculate risks, which may be particularly useful in certain use cases. In general terms, the risk assessment component may be integrated as part of the detection module or could be deployed completely independently. Bearing in mind the taxonomy of ISRA approaches [20], it is recommended that its development considers qualitative assessment criteria, service-based perspectives, vertical valuation and propagated measurement.

5.1.3. Asset Inventory

The asset inventory builds and manages a list of resources or assets to be considered at the risk valuation step. Due to the ability of the 5G mobile networks to automate the deployment of new services and network devices depending on their status, this component plays a critical role in the analysis of the gathered information. The new assets are detected at the monitoring layer and are reported by discovery events. When the existing events are updated, the monitor layer emits modification or removal events. For the proper functioning of the proposed architecture, it is very important to ensure coherence between the list of assets and the real network resources.

5.1.4. Risk Map

This component builds and manages a risk map of the network incidences considering aggregated high-level metrics, events and the inferred risks. The risk map is mainly considered in the following situations: prediction of threats spreading, the establishment of quarantine regions, identification of the best spots to deploy mitigation actions, and recognition of the source of the attacks. As in the case of the asset inventory, there must be coherence between the list of assets and the current network resources. The risk map is built considering the network map, so its proper development implies upgrades in real-time of the network connectivity and its status (throughput, congestion, transmission delays, availability, etc.)

5.1.5. Prediction

Prediction facilitates the anticipation of future complications, such as congestion of certain network regions, inclusion of large amounts of new assets or spreading of cyberattacks. The general scheme for forecasting SA studies variations on their contents by modeling the sequence of observations as time series, as it is shown in Figure 4. Each observation includes information about every SA feature, such as the network map, risk levels or recent incidents monitored over a period of time. Because of this, the prediction module is not only capable of anticipating particular situations, but also the whole future SA, thus providing general and particular overviews about everything that occurs in the system. Note that the observations are delimited by a fixed period of time. A common alternative to this fixed value is to consider more specific features, such as workload or number of situations reported. Both cases imply advantages and disadvantages, but the delimitation by time periods poses a more intuitive method. In order to avoid overloading of storage systems, after a reasonable period, the oldest observations are discarded to make way for the new ones. In this way, a sliding window of size N that gathers the observations is applied. This boundedness is important for ensuring that the implemented algorithms are computable, avoiding the case $N \rightarrow \infty$ (i.e., the worst case), which leads to holding an infinite amount of information.

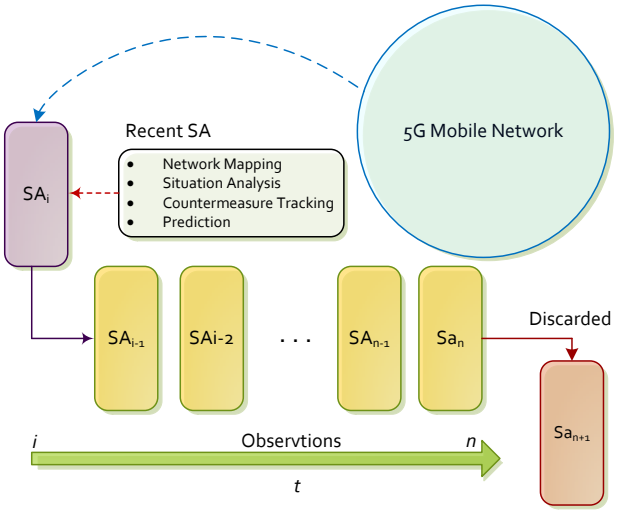

Figure 4. Situational Awareness prediction.

5.1.6. Diagnosis

Diagnosis performs advanced analysis of risks and their assessment, impact, projection and network status. This allows for identifying complex situations that can be difficult to detect from the lower levels of data processing. For example, the diagnosis component should be able to recognize botnets by analyzing the relationships between the risk in network devices compromised by malware and the discovery of surrounding traffic anomalies. Given that the proposed architecture assumes a service-driven vertical risk analysis model that considers propagation, it is important to determine two diagnosis criteria: particular threat level and propagated threat level. The particular threat level related with a risk is its severity. This value is useful to manage isolated situations, but it does not take into account what happens in the surrounding area. On the other hand, the propagated threat level considers all the risks detected in a region and their relationships. There are several publications that address the calculation of propagated risks, where the Bayesian Networks (BN) are the most widespread solutions [44,45]. An example of this methodology is shown in Figure 5, where several particular threat levels related to each other are normalized and pooled, thus allowing for inferring the propagated risk in the lower situations.

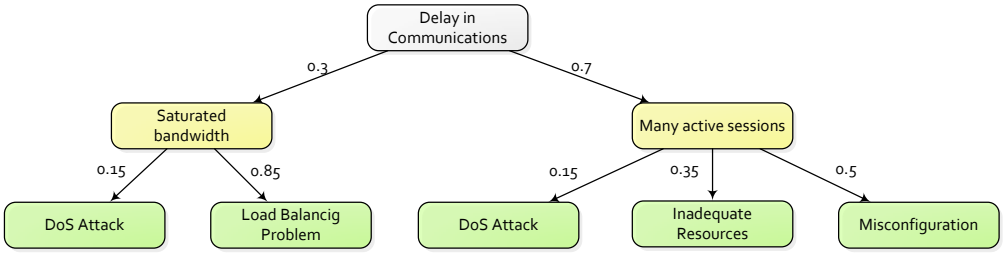

Figure 5. Example of Bayesian Network (Decision Tree) in network diagnosis.

5.1.7. Countermeasure Tracking

The countermeasure tracking component conducts comprehensive monitoring of the actions proposed by the decision-making modules for dealing with situations. This allows for identifying ineffective countermeasures that may lead to new diagnosis or prevention of counterproductive situations. On the other hand, the countermeasure tracking stage allows for the development of an immune memory. Therein, all situations that have been resolved or are being processed are stored. This allows knowing how a problem was previously solved, offering added value to the decision-making. In addition, it provides information about all similar problems that are being processed, which may facilitate the correlation of incidences. In order to enhance the countermeasure tracking tasks, the proposed architecture implements an Issue Tracking System (ITS). The ITS assigns a ticket to every situation tracked in the protected environment. Tickets are running reports on a particular problem, which contains their nature, history and other relevant data. They are continuously analyzed to provide real-time status of the situation, and they record all the countermeasures implemented over time, as well as their effectiveness. The ITS is illustrated in Figure 6, where the original situation to be treated is provided by a use case, or it is detected from the network map. Then, the incident is analyzed, and a token with the results is sent to the decision stages. If countermeasures should not be applied, the ticket is close; in this case, it is assumed that the problem is fixed, and the solution is stored in the immune memory. Otherwise, countermeasures are applied. Then, the evolution and effectiveness of such actions are studied and added to the ticket history field; as in the previous case, the progress is also stored in the memory. The resulting ticket is sent back to the decision-making module, and this process is repeated until the problem is fixed.

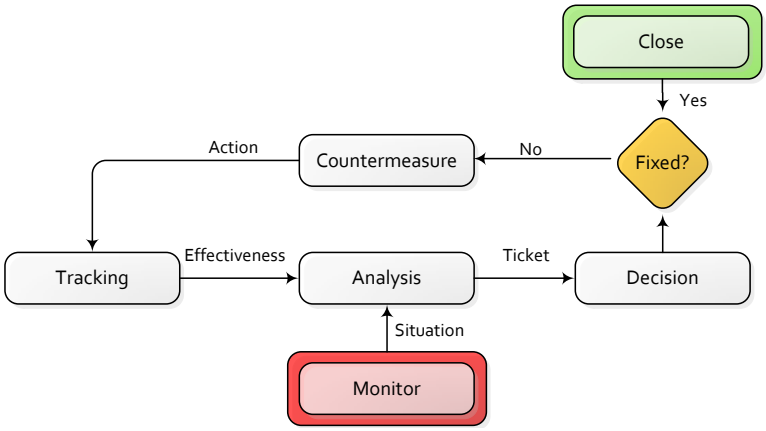

Figure 6. Issue tracking algorithm.

5.2. Decision-Making and Actuators

The decision-making component addresses the problem of mitigating the network situations that may disrupt the normal operation of the network elements and the services provided. In particular cases, it can take also part in the performance optimization process of the services offered by the mobile

network, thereby playing an active role in tasks such as load balancing or management of traffic with multimedia content. To this end, the decision-making processes receive information from the analysis stage (mainly from diagnosis and countermeasures tracking) and then select a set of responses to be executed. According to ISO/IEC 7498-2 [46], such correctives are referred as security safeguards, and they are involved in prevention, mitigation and source identification. The strategy for selecting optimal countermeasures must balance the cost of implementation of the safeguard and the achieved reduction of the incidence impact [47]. The available actions are distributed as NFV-Apps, so that these constitute a large repository of potential countermeasures. The network components that execute the safeguards are known as actuators. An example of their implementation is the mitigation of distributed denial of service attacks: when the analysis component performs a diagnosis related to this category of incidences, the report includes information about the compromised assets, impact, prediction and the attack vector. In the example, the first step of decision-making is to prevent the spreading of the threat by deploying firewall NFV-Apps (specific actuators), thus taking into account the predictions of their distribution. The second step is to mitigate the threat by implementing honeypot NFV-Apps and the malicious traffic redirection towards sinkholes. Once the impact of the attack is minimized, the last step is identifying the sources by applying IP traceback algorithms [48], from which they could be blocked or alerted. The deployment of the various actions is coordinated by an orchestrator agent, which ensures that the virtual resources that implement the countermeasures are available and do not affect the system performance.

6. Conclusions

The conventional schemes for Information Security Risk Management have demonstrated significant shortcomings in the deployment in dynamic monitoring environments, as is the case of 5G mobile networks. In order to contribute to their development, this article has presented a novel architecture for incidence management in 5G based on the combination of the cognitive model for Situational Awareness proposed by Endsley, and the guidelines, platforms and more frequent regulations on the identification and assessment of threats. In this way, the automation of proactive/reactive deployment of countermeasures is facilitated.

Additionally, other important aspects are enhanced such as taking advantage of every information source, the quality of the context to be considered in the decision-making and the projection of the system status. Our design significantly reduces capital and operational expenditures, and it also allows for processing information gathered from 5G mobile networks, expressed as high-level metrics (HoN). The proposed architecture is possible thanks to the capabilities offered by innovative technologies such as SDN, NFV or virtualization. However, in order to not add additional complexity to this first approach, several aspects (to bear in mind before its implementation) have not been explained in detail. This is the case of the characteristics of sensors/actuators and their relationship with the repositories of NFV-Apps from which they can be instantiated.

The paper has not delved into the advanced diagnostic methods to be considered in order to take advantage of the issue tracking system, nor in their feedback with decision-making. The same happened with the specification of the interfaces that connect the different components, and the tactical language for information exchange. All of these aspects will be covered in future work. From the presented work, different lines of research have been encouraged. The most obvious is focused on the implementation of the approach in recent monitoring environments, as occurs with the European project that is funding this investigation.

Another aim could be to analyze their different use cases in order to adapt the incidence concept to common problems related to the Quality of Experience, such as the prevention of internal network errors, collisions or the optimization of multimedia content transmissions. Finally, this architecture is proposed as an alternative to the current collaborative defensive strategies due its great potential for synchronizing the efforts of prevention, detection, mitigation and source identification.

Acknowledgments: This work is supported by the European Commission Horizon 2020 Programme under grant agreement number H2020-ICT-2014-2/671672—SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks). Lorena Isabel Barona López and Ángel Leonardo Valdivieso Caraguay are supported by the Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación SENESCYT (Quito, Ecuador) under the Convocatoria Abierta 2012 and 2013 Scholarship Program.

Author Contributions: The authors contributed equally to this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Andrews, J.G.; Buzzi, S.; Choi, W.; Hanly, S.V.; Lozano, A.; Soong, A.C.K.; Zhang, J.C. What Will 5G Be? *IEEE J. Sel. Areas Commun.* **2014**, *32*, 1065–1082.
2. Agyapong, P.K.; Iwamura, M.; Staehle, D.; Kiess, W.; Benjebbour, A. Design Considerations for a 5G Network Architecture. *IEEE Commun. Mag.* **2014**, *52*, 65–75.
3. NGMN Alliance. NMGN 5G White Paper. 2015. Available online: https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf (accessed on 19 December 2016).
4. Panwar, N.; Sharma, S.; Singh, A.K. A Survey on 5G: The Next Generation of Mobile Communication. *Phys. Commun.* **2016**, *18*, 64–84.
5. Boccardi, F.; Heath, R.W.; Lozano, A.; Marzetta, T.; Popovski, P. Five Disruptive Technology Directions for 5G. *IEEE Commun. Mag.* **2014**, *52*, 74–80.
6. Imran, A.; Zoha, A. Challenges in 5G: How to Empower SON with Big Data for Enabling 5G. *IEEE Netw.* **2014**, *28*, 27–33.
7. International Organization for Standardization and the International Electrotechnical Commission. ISO/IEC 27002: Information Technology, Security Techniques, Code of Practice for Information Security Management. 2005. Available online: http://www.iso.org/iso/catalogue_detail?csnumber=54533 (accessed on 19 December 2016).
8. National Institute of Standards and Technology. NIST-SP800 Series Special Publications on Computer Security. Available online: http://csrc.nist.gov/publications/PubsSPs.html#SP800 (accessed on 19 December 2016).
9. Forum of Incident Response and Security Teams. CVSS: Common Vulnerability Scoring System. Available online: https://www.first.org/cvss/specification-document (accessed on 19 December 2016).
10. MAGERIT: Risk Analysis and Management Methodology for Information Systems. Available online: http://administracionelectronica.gob.es/pae_Home/pae_Documentacion/pae_Metodolog/ (accessed on 19 December 2016).
11. Parvizi, R.; Oghbaei, F.; Khayami, S.R. Using COBIT and ITIL Frameworks to Establish the Alignment of Business and IT Organizations as One of the Critical Success Factors in ERP Implementation. In Proceedings of the 5th IEEE Conference on Information and Knowledge Technology (IKT), Shiraz, Iran, 28–30 May 2013; pp. 274–278.
12. Webb, J.; Ahmad, A.; Maynard, S.B.; Shanks, G.; Popovski, P. A Situation Awareness Model for Information Security Risk Management. *Comput. Secur.* **2014**, *44*, 1–15.
13. Endsley, N.R. Design and Evaluation for Situation Awareness Enhancement. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Anaheim, CA, USA, 24–28 October 1988; Volume 32, pp. 97–101.
14. Leau, Y.B.; Ahmad, A.; Manickam, S. Network Security Situation Prediction: A Review and Discussion. In Proceedings of the 4th International Conference on Soft Computing, Intelligent Systems, and Information Technology (ICSIIT), Bali, Indonesia, 11–14 March 2015; pp. 424–435.
15. Marquezan, C.C.; Mahmood, K.; Zafeiropoulos, A.; Krishna, R.; Huang, X.; An, X; Corujo, D; Leitão, F.; Rosas, M.L.; Einsiedler, H. Context Awareness in Next Generation of Mobile Core Networks. *arXiv* **2016**, arXiv:1611.05353.
16. Belmonte Martin, A.; Marinos, L.; Rekleitis, E.; Spanoudakis, G.; Petroulakis, N.E. *Threat Landscape and Good Practice Guide for Software Defined Networks/5G*; European Union Agency for Network and Information Security (ENISA): Heraklion, Greece, 2015. Available online: http://openaccess.city.ac.uk/15504/7/SDN%20Threat%20Landscape.pdf (accessed on 19 December 2016).

17. 5G-Ensure Project. Enablers for Network and System Security and Resilience. Project Reference: 671562. Funded under: H2020-ICT-2014-2. Available online: http://www.5gensure.eu/ (accessed on 19 December 2016).
18. 5G Ensure. Deliverable D 2.3, Risk Assessment, Mitigation and Requirements (Draft), August 2016. Available online: http://www.5gensure.eu/deliverables (accessed on 19 December 2016).
19. Aven, T. Risk Assessment and Risk Management: Review of Recent Advances on their Foundation. *Eur. J. Oper. Res.* **2016**, *256*, 1–13.
20. Shameli-Sendi, A.; Aghababaei-Barzegar, R.; Cheriet, M. Taxonomy of Information Security Risk Assessment (ISRA). *Comput. Secur.* **2016**, *57*, 14–30.
21. Hansson, S.O.; Aven, T. Is Risk Analysis Scientific? *Risk Anal.* **2014**, *34*, 1173–1183.
22. Doty, P. US Homeland Security and Risk Assessment. *Gov. Inf. Q.* **2015**, *32*, 342–352.
23. Yang, M.; Khan, F.; Lye, L.; Amyotte, P. Risk Assessment of Rare Events. *Process Saf. Environ. Prot.* **2015**, *98*, 102–108.
24. Ab Rahman, N.H.; Choo, K.K.R. A Survey of Information Security Incident Handling in the Cloud. *Comput. Secur.* **2015**, *49*, 45–69.
25. Knowles, W.; Prince, D.; Hutchison, D.; Disso, J.; Ferdinand, P.J.; Jones, K. A Survey of Cyber Security Management in Industrial Control Systems. *Int. J. Crit. Infrastruct. Prot.* **2015**, *9*, 52–80.
26. Ni, S.; Zhuang, Y.; Gu, J.; Huo, Y. A Formal Model and Risk Assessment Method for Security-critical Real-time Embedded Systems. *Comput. Secur.* **2016**, *58*, 199–215.
27. Cherdantseva, Y.; Burnap, P.; Blyth, A.; Eden, P.; Jones, K.; Soulsby, H.; Stoddart, K. A Review of Cyber Security Risk Assessment Methods for SCADA Systems. *Comput. Secur.* **2016**, *56*, 1–27.
28. Quick, D; Martini, B; Choo, K.K.R. Cloud Storage Forensics. In *Syngress*; Elsevier: Amsterdam, The Netherlands, 2013; pp. 1–208, ISBN: 978-0-12-419970-5. Available online: http://www.sciencedirect.com/science/book/9780124199705 (accessed on 19 December 2016).
29. Ab Rahman, N.H.; Glisson, W.B.; Yang, Y.; Choo, K.K.R. Forensic-by-Design Framework for Cyber-Physical Cloud Systems. *IEEE Cloud Comput.* **2016**, *3*, 50–59.
30. Ab Rahman, N.H.; Cahyani, N.D.W.; Choo, K.K.R. Cloud incident handling and Forensic-by-Design: Cloud Storage as a Case Study. *Concurr. Comput. Pract. Exp.* **2016**, 1–16, doi:10.1002/cpe.3868.
31. Endsley, M.R.; Selcon, S.J.; Hardiman, T.D.; Croft, D.G. A Comparative Analysis of SAGAT and SART for Evaluations of Situation Awareness. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Chicago, IL, USA, 5–9 October 1998; Volume 1, pp. 82–86.
32. Adam, E.C. Fighter Cockpits of the Future. In Proceedings of the 12th IEEE Digital Avionics Systems Conference (DASC), Fort Worth, TX, USA, 25–28 October 1993; pp. 318–323.
33. Dahal, N.; Abuomar, O.; King, R.; Madani, V. Event Stream Processing for Improved Situational Awareness in the Smart Grid. *Expert Syst. Appl.* **2015**, *42*, 6853–6863.
34. Naderpour, M.; Nazir, S.; Lu, J. The Role of Situation Awareness in Accidents of Large-scale Technological Systems. *Process Saf. Environ. Prot.* **2015**, *97*, 13–24.
35. Moradi-Pari, E.; Tahmasbi-Sarvestani, A.; Fallah, Y.P. A Hybrid Systems Approach to Modeling Real-time Situation-Awareness Component of Networked Crash Avoidance Systems. *IEEE Syst. J.* **2016**, *10*, 169–178.
36. Seppänen, H.; Virrantaus, K. Shared Situational Awareness and Information Quality in Disaster Management. *Saf. Sci.* **2015**, *77*, 112–122.
37. Chatzimichailidou, M.K.; Stanton, N.A.; Dokas, I.M. The Concept of Risk Situation Awareness Provision: Towards a New Approach for Assessing the DSA about the Threats and Vulnerabilities of Complex Socio-technical Systems. *Saf. Sci.* **2015**, *79*, 126–138.
38. Franke, U.; Brynielsson, J. Cyber Situational Awareness—A Systematic Review of the Literature. *Comput. Secur.* **2014**, *46*, 18–31.
39. Schmittling, R.; Munns, A. Performing a Security Risk Assessment. *ISACA J.* **2010**, *1*, 10–18.
40. Naderpour, M.; Lu, J.; Zhang, G. A Situation Risk Awareness Approach for Process Systems Safety. *Saf. Sci.* **2014**, *64*, 173–189.
41. Tahaei, H.; Salleh, R.; Khan, S.; Izard, R.; Choo, K.K.R.; Anuar, N.B. A multi-objective Software Defined Network Traffic Measurement. *Measurement* **2016**, *95*, 317–327.
42. Shanbhag, S.; Kandoor, A.R.; Wang, C.; Mettu, R.; Wolf, T. VHub: Single-stage Virtual Network Mapping Through Hub Location. *Comput. Netw.* **2015**, *77*, 169–180.

43. Chowdhury, N.M.K.; Rahman, M.R.; Boutaba, R. Virtual Network Embedding with Coordinated Node and Link Mapping. In Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM), Rio de Janeiro, Brasil, 19–25 April 2009; pp. 783–791.
44. Awan, M.S.K.; Burnap, P.; Rana, O. Identifying Cyber Risk Hotspots: A Framework for Measuring Temporal Variance in Computer Network Risk. *Comput. Secur.* **2016**, *57*, 31–46.
45. Shin, J.; Son, H.; Ur, R.K.; Heo, G. Development of a Cyber Security Risk Model Using Bayesian Networks. *Reliab. Eng. Syst. Saf.* **2015**, *134*, 208–217.
46. International Organization for Standardization and the International Electrotechnical Commission. ISO/IEC 7498-2, Information Processing Systems—Open Systems Interconnection—Basic Reference Model Part 2: Security Architecture. Available online: http://www.iso.org/iso/catalogue_detail.htm?csnumber=14256 (accessed on 19 December 2016).
47. Gonzalez-Granadillo, G.; Garcia-Alfaro, J.; Alvarez, E.; El-Barbori, M.; Debar, H. Selecting Optimal Countermeasures for Attacks against Critical Systems Using the Attack Volume Model and the RORI Index. *Comput. Electr. Eng.* **2015**, *47*, 13–34.
48. Alenezi, M.N.; Reed, M.J. Uniform DoS Traceback. *Comput. Secur.* **2014**, *45*, 17–26.

entropy

MDPI

Article

An Approach to Data Analysis in 5G Networks

Lorena Isabel Barona López †, Jorge Maestre Vidal † and Luis Javier García Villalba *,†

Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), Faculty of Computer Science and Engineering, Office 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases, 9, Ciudad Universitaria, Madrid 28040, Spain; lorebaro@ucm.es (L.I.B.L.); jmaestre@ucm.es (J.M.V.)

* Correspondence: javiergv@fdi.ucm.es; Tel.:+34-91-394-7638

† These authors contributed equally to this work.

Academic Editor: Kevin H. Knuth

Received: 16 January 2017; Accepted: 14 February 2017; Published: 16 February 2017

Abstract: 5G networks expect to provide significant advances in network management compared to traditional mobile infrastructures by leveraging intelligence capabilities such as data analysis, prediction, pattern recognition and artificial intelligence. The key idea behind these actions is to facilitate the decision-making process in order to solve or mitigate common network problems in a dynamic and proactive way. In this context, this paper presents the design of Self-Organized Network Management in Virtualized and Software Defined Networks (SELFNET) Analyzer Module, which main objective is to identify suspicious or unexpected situations based on metrics provided by different network components and sensors. The SELFNET Analyzer Module provides a modular architecture driven by use cases where analytic functions can be easily extended. This paper also proposes the data specification to define the data inputs to be taking into account in diagnosis process. This data specification has been implemented with different use cases within SELFNET Project, proving its effectiveness.

Keywords: 5G; data analysis; network function virtualization; situational awareness; software defined networking

1. Introduction

5G networks expect to provide a secure, reliable and high-performance environment with minimal disruptions in the provisioning of advanced network services, regardless the device location or when the service is required [1]. This new network generation will be able to deliver ultra-high capacity, low latency and better Quality of Service (QoS) compared with current Long Term Evolution (LTE) networks [2]. In order to provide these capabilities, 5G proposes the combination of advanced technologies such as Software Defined Networking (SDN) [3,4], Network Function Virtualization (NFV) [5,6], Cloud Computing [7], Self-organized Networks (SON) [8,9], Artificial Intelligence, Big Data [10–12], Device to Device Communications (D2D), among others [13–17]. In particular, 5G will be able to face unexpected changes or network problems through the identification of specific situations and taking into account the user needs and the Service level Agreements (SLAs).

Nowadays, the main telecommunication operators and community research are working in strategies to facilitate the decision-making process when specific events or situations compromises the health in 5G Networks [18,19]. Meanwhile, the concept of situational awareness (SA) and incident management models applied to 5G Networks are also an emerge topic [20,21]. In this context, Self-Organized Network Management in Virtualized and Software Defined Networks Project (SELFNET) [22] combines SDN, NFV and SON concepts to provide a smart autonomic management framework, analysing and resolving network problems and improving the QoS and Quality of

Experience (QoE) of end users. In order to facilitate the decision-making process, SELFNET proposes an analysis phase to diagnosis and predict possible problems in 5G Networks.

This paper presents the design of SELFNET Analyzer Module, which main objective is to diagnose the network state and infer data from monitored low level and aggregated metrics in order to facilitate proactive responses. The contributions of this proposal include: (i) the description of diagnosis and prediction capabilities in 5G environments and how it is being applied in current research and projects; (ii) the introduction of SELFNET Analyzer Architecture, its design principles and requirements and (iii) the definition of data specification, as well examples, to obtain the initial parameters to diagnostic purpose. This document is organized into eight sections, being the first of them the present introduction. Section 2 describes 5G requirements, related works, the main characteristics and analytic capabilities of SELFNET Project. Section 3 outlines the design principles, requirements and the architecture of Analyzer Module as a whole. Then, Section 4 shows this module as a black, emphasizing the data inputs and outputs. Section 5 formally defines how the data must be specified. Section 6 illustrates examples of the data specification and their workflows. Section 7 discusses the main contributions of this proposal. Finally, conclusion and future work are presented in Section 8.

2. Background

This section describes how analysis and intelligent capabilities can address 5G requirements, the related work and research projects, emphasizing the main features of SELFNET Project.

2.1. Diagnosis Capabilities in 5G Networks

A 5G network envisages an architecture able to cover three main domains [1]: (i) enhancement of radio capabilities to enable the spectrum optimization, the interference coordination and cost-effective dense deployments; (ii) provisioning of an effective network management environment to create and deploy a common core to support several use cases in a cost-effective manner; and (iii) simplification of the system operations by means of automated procedures, where the introduction of new capabilities or network functions should not imply increased complexity on operations and management tasks. In order to tackle these requirements, 5G networks take advantage of the separation between data and control plane (network programmability) offered by SDN architectures [23], the deployment of virtualized network functions, the scalability and flexibility in the service provisioning based on cloud environments, enabling high capacity and massive communications (cognitive radio, carrier aggregation, Machine to Machine Communication), spectrum and resource optimization (millimeter wave and massive Multiple Input Multiple Output (MIMO)) and intelligent capabilities provided by artificial intelligence or self-organization concepts [24,25].

In particular, the introduction of analysis and intelligent capabilities [8,19] could be applied to several domains such as autonomic network maintenance, automation in the provisioning of services, prediction and remediation of congestion or queue utilization, detection of security threats, improving network efficiency, multi cell coordination, provisioning of high QoS and QoE for services, etc. For this purpose, analysis and intelligent capabilities allow to response to network problems based on pattern recognition, the dynamic smart selection of the best location where the services can be deployed or migrated, sharing and releasing of resources based on forecasting methods, building of context awareness models based on real time information from the network, its devices and applications. In order to provide intelligence and facilitate the decision-making process, some tasks must be performed. On one hand, analysis stage is intended to perform the identification of network situations and events. These situations do not necessarily imply (a priori) a harmful nature. On the other hand, the decision-making task determines if a specific situation is a risk for the network health, or its components, and then it performs the respective countermeasures.

In this context, traditional approaches apply different analysis and reasoning techniques, such as Bayesian Networks (BN) [26], in order to provide intelligent to common network management tasks. However, these models are not sufficient to guarantee the network performance according to SLAs and

future 5G user needs [1]. There are some proposals to address the data analysis in 5G systems and its elements such as access and radio components [13,27], network devices [28], cloud elements [29] and resource allocation [30]. In [31], a prototype to perform mobile network analysis based on Markov Logic Network (MLN) and semantic web technologies is presented. This approach allows the optimization and network status characterization but does not explain how it cover heterogeneous data sources. For its part, Imran et al. [10] proposes a framework to provide a full view of network status based on machine learning and big data concepts. To this end, their proposal predicts the user behaviour and dynamically associate the network response to the network parameters. However it is doesn't specify how to deal with SDN or NFV components.

Meanwhile, there are reports [1,32] and projects [21,33–37] that introduce analysis capacities to cover 5G requirements. In this way, METIS Project [28] takes into account SON concept in order to provide a new level of adaptability to 5G infrastructures. Meanwhile, 5G-NORMA [35] introduces adaptive capacities to allocate network functions based on user and traffic demands over time and location. CHARISMA project [36] deploys an intelligent cloud radio access network and end devices. For its part, 5G-Ensure [21] proposes a 5G secure system based on risk assessment and mitigation methodologies. COGNET [37] takes into account Machine learning, SDN and NFV to provide dynamic adaptation of network resources. For its part, a whole approach to address not only analysis component but also the whole cycle of incident management in 5G networks is proposed in [20]. This work applies the three stages of processing information of Endsley Model [38] to 5G Networks: perception, comprehension and projection. In the perception phase the monitoring and collection of different metrics from network infrastructure (and its elements) are performed. Then in comprehension stage, the association and correlation of this information are performed in order to provide enhanced metrics to be analysed (projection phase). The analysis component includes the diagnosis and prediction of the whole state of the system. In general terms, these proposals aid to tackle 5G Requirements but they do not offer a generalized approach able to take into account several kind of metrics from heterogeneous data sources, that is the case of SELFNET Project [22].

2.2. SELFNET Project

The SELFNET H2020 Project [22] aims to provide an autonomic network management framework for 5G mobile network infrastructures through the integration of novel technologies such as SDN, NFV, SON, Cloud computing and Artificial Intelligence. SELFNET enables both autonomic corrective and preventive actions to mitigate existing or potential network problems while providing scalability, extensibility and reduce capital expenditure (capex) and operational expenditure (opex). These capabilities are provided through a layered architecture and a use-case driven approach, as is detailed in [34]. The SELFNET architecture addresses major network management problems including self-protection capabilities against distributed cyber-attacks, self-healing capabilities against network failures, and self-optimization to dynamically improve the performance of the network and the QoE of the users. For this purpose, SELFNET defines two kind of advanced network functions: (i) sensors to monitor specific information from the network and (ii) actuators to address or mitigate possible problems. In particular, the network intelligence is provided by SON Autonomic Layer. This layer collects metrics related with the network behaviour and use that information to infer the network status. Then, it decides the actions to be executed to accomplish the system goals. The SON Autonomic Layer is composed by two sublayers: (i) Monitor and Analyzer Sublayer and (ii) Autonomic Management Sublayer. The Monitor and Analyzer Sublayer follows the Endsley Situational Awareness Principles. Monitoring and Discovery, Aggregation and Correlation and Analyzer modules corresponds with the Perception, Comprehension and Projection functions as is shown in Figure 1.

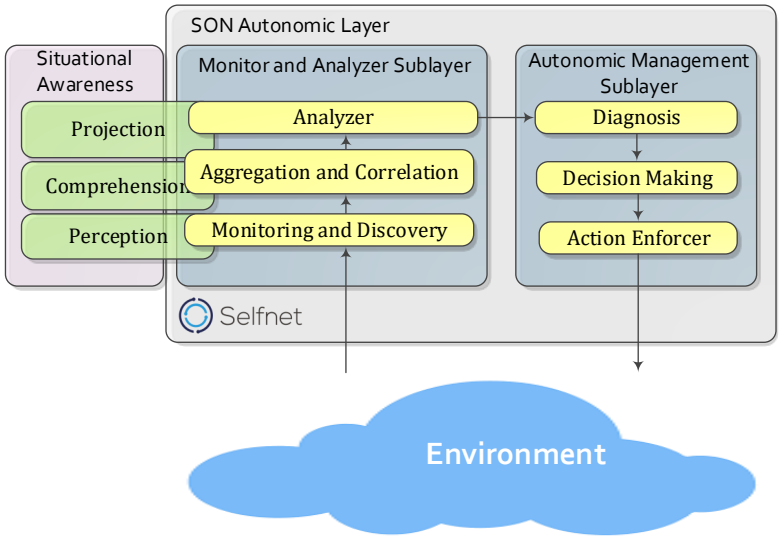

Figure 1. Endsley vs. SELFNET Autonomic Layer.

Regarding the Analyzer Module, its main goal is to infer data from the monitored metrics in order to facilitate proactive responses over the network infrastructure (i.e., enhance diagnosis and decision-making tasks). Therefore the Analyzer Module is the first step to provide intelligence to the system, where complex conclusions should be reached by reasoning about knowledge provided by the Monitoring/Aggregation stages and the definition of each use case. Because of this, the Analyzer Module distinguishes three great information processing activities: Pattern Recognition, Reasoning and Prediction. The achieved conclusions are described in the form of symptoms related with each use case. Bearing this in mind, it is possible to assert that the Analyzer Module provides a symptom-oriented Situational Awareness bounded by the situations defined for each use case.

3. SELFNET Analyzer Module Design

In this section the design of SELFNET Analyzer Module is detailed. It also describes the initial assumptions, the requirements, the design principles as well as the Analyzer architecture.

3.1. Initial Assumption and Requirements

The following describes the most relevant requirements and the main initial assumptions considered in the design of the Analyzer Module:

- *Scalability*. The approach must be allowed to add new capabilities (extensibility), according to SELFNET design principles. For this reason, the integration of additional analytic functionalities are done via plugins.
- *Use Case Driven*. Given the heavy reliance of the tasks performed with the characteristics of use cases, the basic definition of the observations to be studied (Knowledge-based objects, rules, prediction metrics, etc.) are provided by the use case operator, thus being the Analyzer Module scalable to alternative contexts.
- *Knowledge Acquisition*. It is well known that the most common disadvantage of the expert systems is the initial knowledge acquisition problem. Hence, to have skilled operators in novel use cases with the ability to properly specify rules is not always straightforward. This document does not address the issue of the innate knowledge acquisition. Our approach assumes that the use case knowledge-bases are provided by skilled operators or by accurate machine learning algorithms.
- *User-Friendly Symptom Definition Rules*. The definition of proper rule-sets is a tricky business. Thus, even the skilled operators often do coherence/ambiguity mistakes. In order to mitigate these

problems, the configuration and definition of new use cases should be user-friendly, as well as the scheme for building new rule-sets.

- *Uncertainly*. Classical logic permits only exact reasoning. It assumes that perfect knowledge always exists, but this remains far from the SELFNET reality. In order to improve the quality of the conclusions, the Analyzer Module manages the knowledge bearing in mind uncertainty. This is particularly appropriate for certain analytic features, such as studying observations based on decision thresholds or confidence intervals. In addition, closing the door on possible stochastic dependent definitions is against the SELFNET design principles, as these could be the keys to properly specify future use cases.
- *Filtering*. Initially, the filtering of symptom reports is not considered. Because of this, every inferred symptom, regardless of nature or uncertainty, is transmitted to the diagnosis/decision-making stages, where their impact and relevance are properly assessed.

3.2. Design Principles

The following design principles and limitations lay the foundation of the Analyzer framework, as well as the implementation of its internal components:

- *Big Data*. In order to deal with huge and homogeneous datasets, Big Data provides predictive algorithms, user behaviour analytics, and aggregation/correlation functionalities [39]. These capabilities are mainly taken into account in monitoring and aggregation tasks. The Analyzer Module deals with aggregated and correlated metrics, hence reducing the amount of information to be analysed. In our approach, the implementation of Big Data technologies to handle all this information is optional, leaving the decision of integrate these tools at the mercy of the SELFNET administrators, which driven by a better awareness of the use cases and the monitoring environment are more able to decide whether they are counterproductive or beneficial [40]. Because of this, our contribution is compatible with both Big Data and conventional techniques.
- *Stationary Monitoring Environment*. According to Holte et al. [41], in a stationary monitoring environment, the characteristics and distribution of the normal observations to be analysed match the reference sample population considered in the Analyzer learning processes. If the monitoring environment distribution is able to change representatively, it is considered non-stationary. Another problem that may reduce the quality of the analytics is the presence of gradual changes over time in the statistical characteristics of the class to which an observation belongs. In the literature this fluctuation is known as concept-drift. These problems are discussed at length in [42]. The assumption that the Analyzer Module operates on a stationary environment brings a simple and efficient solution, but prone to slight failures when the changes occur. On the other hand, to consider a non-stationary monitoring environment improves accuracy, but entails new challenges, among them: detection of changes, implementation of model/regression updating techniques, identifying when the calibration must be completed or selection of the samples that will be taken into account in new trainings. Given the complexity that this implies, the Analyzer Module assumes a stationary monitoring environment. The non-stationary approach will be part of future work.
- *High Dimensional Data*. The analysis of high dimensional data implies to bear in mind data whose dimension is larger than dimensions considered in classical multivariate analysis. As indicated by Bouveyron et al. [43], when conventional methods deal with high dimensional data they are susceptible to suffer the well-known curse of dimensionality, where considering a large number of irrelevant, redundant and noisy attributes leads to important prediction errors. Hence operate with this data implies the need for more specific and complex algorithms. In terms of SELFNET this means that the vector of Health of Network Metrics (HoN) is large enough to consider the implementation of specific methods adapted to optimize the processing tasks of this kind of information. A priori there are no signs of SELFNET requiring processing an important amount of High Dimensional Data. Therefore, this paper does not take into account differences between

conventional and High Dimensional data, assuming that the aggregation tasks will be able to optimize the amount of attributes to be analysed.

- *Supervision*. SELFNET training mode. The analytical methods based on modeling/regression assume that new knowledge can be inferred from observations, by a prior learning stage. The learning process often requires reference data which allows identifying the most characteristic features of the monitoring environment, such as rules, boundaries, incidence matrices, direction vectors or basic statistics. Given the complexity involved in designing a SELFNET training mode, this approach describes how the information needed for the construction of new models is obtained.
- *Centralized Design*. To assume a centralized approach lead us to pose a general purpose scheme where the onboarding of new use cases is completely configurable by specification, and which does not requires updating the implementation (see Figure 2). Therefore the centralized approach is not dependent on the characteristics of the use case, so it is highly scalable and allows performing tasks efficiently (avoiding redundancy). However, its design and the description of use cases is complex. On the other hand, the distributed approach includes an additional component for each use case in which specific pattern recognition and prediction methods are implemented via plugin. The preprocessing, selection and symptom discovery mechanisms have general purpose. In essence, this second approach is easy of design, but completely use case dependent; each time a new use case is onboarded, the Analyzer implementation must be updated. Due to the large impact on the scalability that this entails, the centralized approach is considered hereinafter.

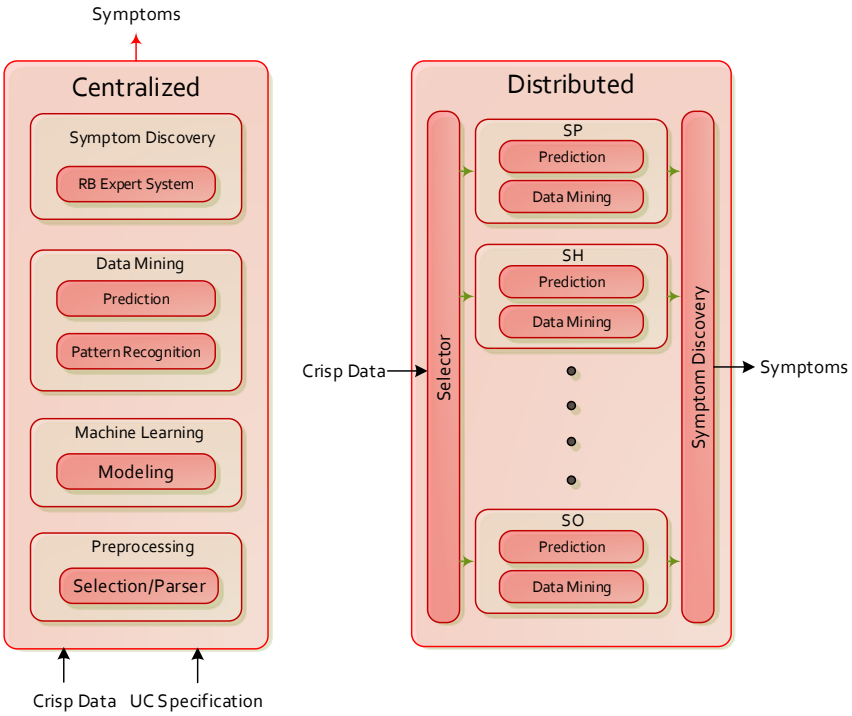

Figure 2. Centralized and Distributed Architectures.

- *Data Encapsulation*. The greatest challenge in designing the SELFNET Analyzer approach is the requirement of dealing with unknown data. It is possible to assume that use cases do not provide clear enough information about the characteristics of the information to be analyzed (in fact, several future use cases are completely unknown). At the specification stage, use cases operators tend to provide good qualitative information about the metrics to consider, but may overlook details about their quantitative nature: data type, domain, range, restrictions, etc., which is what in the first instance, will be considered in the analysis tasks. Furthermore, quantitative information is much use case dependent. In order to subtract relevance to quantitative details (which are the backbone

of the aggregation/correlation tasks), and thus facilitate the incorporation of new use cases by definition of general purpose descriptors, the SELFNET Analyzer is driven by data encapsulated in two levels of abstraction: quantitative and qualitative parameters (see Figure 3). The first one is independent of the use cases, and allows designing a centralized analysis framework valid for any type of data specification. On the other hand the qualitative parameters gather information directly related with SELFNET and the use case to which they belong (metric name, source, location, tenant, etc.). This data is mainly required for aggregation/correlation, diagnosis and decision making.

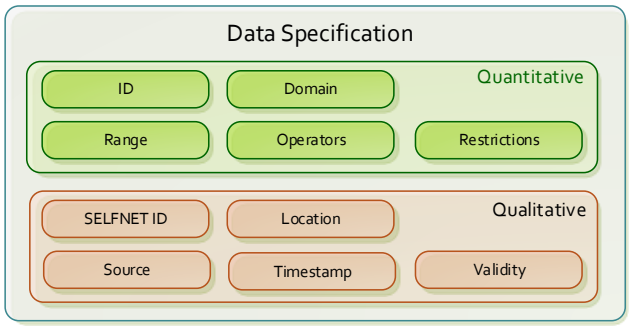

Figure 3. Example of Data Encapsulation.

3.3. Analyzer Module Architecture

In Figure 4 the architecture of the Analyzer Module is illustrated. It is centralized, and distinguishes the following eight core components: (1) Pattern Recognition, (2) Prediction, (3) Adaptive Thresholding, (4) Knowledge-base, (5) Inference Engine, (6) Memory, (7) User Interface and (8) Uncertainty Estimation. The sets (4)–(6),(8) is related with Reasoning, (1)–(3) with Projection and (7) with the administration of the use cases. Their tasks are summarized below:

- *Pattern Recognition*: identifies previously known or acquired patterns and regularities in facts related with aggregate data (i.e., $Fa(T_h)$, $Fa(KPI)$, $Fa(Ev)$), and returns Facts Fa with the results of their study. With this purpose, different internal tasks may be executed: study of the input data (both training data and samples to be analysed), decision of the best suited data mining strategies for each context, feature extraction, construction of models/regressions, analysis of facts related with aggregate data in order to find and labeling verification. Note that the bibliography collects a plethora of pattern recognition methods, which are adapted to the needs of the use cases and to the characteristics of the different monitoring environments [44]. The SELFNET Analyzer focuses on two fundamental actions: the identification of signatures of previously known events [45] and the detection of anomalies [46].
- *Prediction Component*: calculates the prediction metrics (as Facts Fa) associated to each use case from the observations provided by the aggregation stage (Thresholds T_H, Key Performance Indicators KPI and Events Ev). This implies different processing steps: management of a track record with the data required to build forecasting models, analysis of the data characteristics which are relevant for deciding the best suited prediction algorithms, construction of forecasting models, decision of prediction algorithms, forecasting and evaluation of the results in order to learn from the previous decisions. Note that as stated in [47], the prediction of network events enhances the optimization of resources, allows the deployment of proactive actions and anticipates risk identification. The SELFNET Analyzer focuses primarily on infer predictions from two data structures: time series and graphs. The first one aims to determine the evolution of the HoN metrics, hence it mainly implements exponential smoothing algorithms [48] and autoregressive models [49]. On the other hand, the evolution of graphs is predicted in order to anticipate the discovery of new elements [50] and facilitate the management of resources [51].

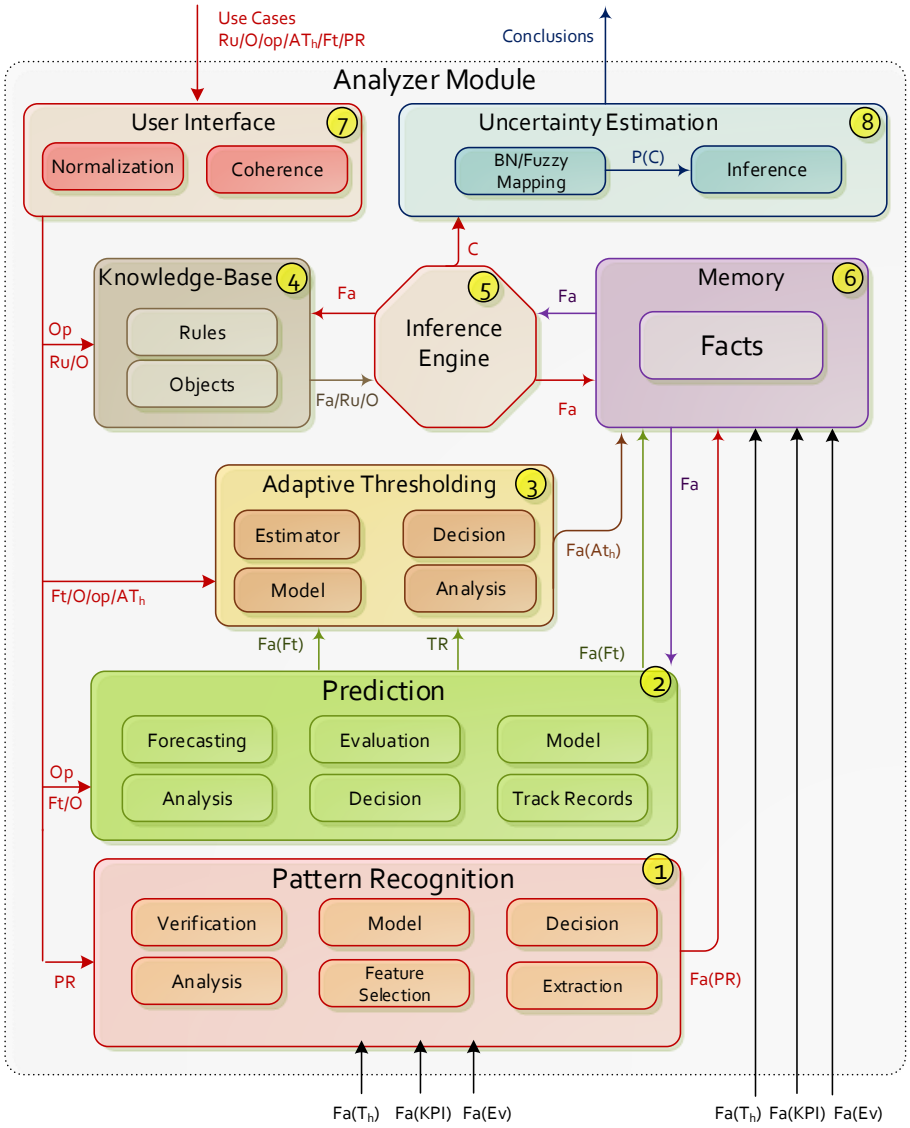

Figure 4. Analyzer Module architecture.

- *Adaptive Thresholding*: establishes measures to approximate when the forecast errors must be taken into account when identifying symptoms. Therefore it receives as input parameters the values related with the prediction metrics (Track Record TR and Forecasts Ft), and returns adaptive thresholds AT_h. Their construction involves different steps, such as analyzing and extracting the main features from the input data, decision of the best suited thresholding algorithms, modeling and estimation of thresholds. The SELFNET Analyzer build adaptive thresholds from data represented as time series or graph, which allows inferring more accurate conclusions from every forecast generated by the prediction component. The main applicability of the adaptive thresholds is considering the context of the monitoring environment in the inference of new facts related with filtering [52], and decreasing the false positives rates [53].
- *Knowledge-Base*: stores specific information about each use case. This data is represented by objects and rules. The objects O are the basic units of information (ex. temperature, congestion, latency, etc.). The rules Ru are the guidelines for reasoning that enable the inference of facts and conclusions. Facts, objects, and their values are interrelated through operations Op. A priori, in this approach online machine learning is not considered in order to acquire knowledge about the use

cases in real-time [54], such as definition of new rules, prioritization, metric weighting, etc. (i.e., all information to be considered part of the original training and the specification of the use cases and their symptoms provided by operators).

- *Inference Engine*: applies rules Ru to the knowledge base in order to deduce new knowledge. This process would iterate as each new fact Fa in the knowledge base could trigger additional rules. Traditionally, inference engines operate in one of two modes: forward chaining and backward chaining [55]. The first initially considers previously known facts and infers new facts. On the other hand, backward chaining initially considers facts and tries to infer the causes that have led to them. Because the SELFNET Analyzer infers conclusions from discovered facts, the first approach is implemented. In addition, it is important to bear in mind that the easier implementation of the inference engine considers basic implication elimination rules (i.e., modus ponens rules) driven by propositional logic [56]. They can be adapted to different representations of uncertainty, such as fuzzy logic [57], rough sets [58] or Bayesian networks [26]. But in order to facilitate the understanding of this proposal, the current specification of rules on the SELFNET Analyzer applies only basic propositional logic rules (as described in Section 5), hence postponing for future works a most complex but generic definition.
- *Memory*: stores all the known facts Fa concerning with the use cases (ex. Temperature $= 3°$, Latency >200 ms, etc.) considering those predicted/inferred $(Fa(PR), Fa(AT_h), Fa(Ft))$ and those provided by the SELFNET Monitoring/Aggregation stages $(Fa(T_h), Fa(KPI), Fa(Ev))$. Metadata related with qualitative additional information about the nature of the discovered facts is also stored.
- *User Interface*: configures Patter Recognition PR for each use case and allows updating the knowledge-base by inserting, modifying or deleting data associated with every use case, such as objects O, rules Ru operations Op or prediction metrics Ft. The information is preprocessed aiming to ensure compatibility and coherence [59]. The latter is particularly important, as it tries to avoid contradictions and ambiguity between rules, prior to their incorporation into the SELFNET intelligence.
- *Uncertainly Estimation*: complements the inference engine and facilitates the study of the conclusions bearing in mind their uncertainty. Its outputs are the acquired conclusions as potential symptoms of relevant incidences, their uncertainty and the information associated with their inference (facts, triggering rules, etc.). This is the only optional element of the architecture, since its use is only required when the SELFNET Diagnostic task [60] need to disambiguate conclusions, filter those of greater uncertainty or convert the logic on the Analyzer to data specified for upper layers of SELFNET. For example, when the inference engine operates on fuzzy logic rules, the element of Uncertainly Estimation generates a quantifiable result use-friendly for Diagnosis as crisp logic, given fuzzy sets and the corresponding membership degrees (i.e., defuzzification) [61].

4. Analyzer Inputs/Outputs

By studying the Analyzer Module as a black box model it is possible to focus more on its inputs/outputs and their relationship with the rest of the SELFNET components [34]. From this perspective, their information sources, nature of the data and behaviour in different circumstances are described. As shown in Figure 5, the Analyzer Module depends on three sources of information. Two of them are external: the SELFNET Aggregation component and the use case operators; the last is internal: data generated by the Analyzer Module itself. The inferred conclusions are reported to the SELFNET Diagnosis Module as symptoms [60]. The role played by each of these elements is detailed below:

- *Aggregation*. Observations in SELFNET come to the Analyzer through the Aggregation Layer (Perception capabilities within the Endsley's model). The information provided by this source contains facts concerning Events $Fa(Ev)$, Thresholds $Fa(T_H)$ and Key Performance Indicators $Fa(KPI)$ related to the current network status.

- *Use Case Operators*. The knowledge-base is specified from data acquired from the use case definitions. Because of this, use case operator may provide inference rules Ru (1), and declare the objects O (2), operations Op (3) and prediction metrics Ft (4) to be taken into account (i.e., what observations should be taken into account (2), how (3)what data must be forecasted (4) and how are they considered in order to acquire knowledge about the specific use case (1)). Optionally, the use case operator may describe the adaptive thresholds AT_h to be calculated, and if pattern recognition PR is required, then configuring how it must be addressed.
- *Analyzer*. An important part of the information necessary for proper Reasoning is generated by the Analyzer Module itself. It is gathered into a pair of groups: Perception and Machine Learning. The first block is imperative, and establishes facts Fa from pattern recognition $Fa(PR)$, forecasts $Fa(Ft)$ and adaptive thresholds $Fa(AT_h)$. On the other hand, machine learning may provide additional data to that provided by use case operators (definition of new rules Ru and description of prediction metrics Ft). Furthermore, it could generate information to improve the knowledge management (weight, prioritizations, fusion, smoothing, etc.).
- *Diagnosis*. the final conclusions and symptoms that compose the SELFNET Situational Awareness are sent to Intelligent Diagnostic Module (Autonomic Management Sublayer) [60], via reports Re.

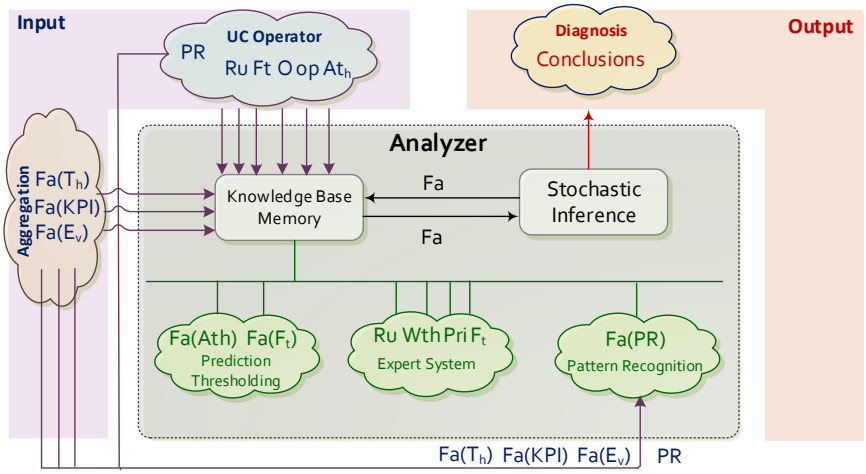

Figure 5. Analyzer Module as a Black Box.

5. Use Case Descriptors

This section describes the characteristics of Analyzer Module quantitative data and its categories. In Table 1 the quantitative data is summarized.

Table 1. Summary of UC data Specification.

Data	Category	Provider	Destination	Format
Object (simple) O	Specification	Use Case	Analyzer	$O_i : \{object \quad name \mid weight \mid noValues \mid range \quad of \quad values \quad Va\}$
Object (mult) O	Specification	Use Case	Analyzer	$O_i : \{object \quad name \mid weight \mid noValues \mid [Va_1][Va_2]...[Va_k]\}$
Operation Op	Specification	Use Case	Analyzer	$Op_i : \{name \mid symbol \mid priority \mid operands \mid description\}$
Facts Fa	Assessment	Agg-Ana	Analyzer	$Fa_i : \{expresion \mid weight \mid uncertainty \mid timestamp \mid location\}$
Rule Ru	Specification	Use Case	Analyzer	$Rule : \{rule \mid priority \mid use \quad case\}$
Forecast (ts) Ft	Specification	Use Case	Analyzer	$Ft_i : \{timeSeries \mid object \mid domain \quad lenght\}$
Forecast (G) Ft	Specification	Use Case	Analyzer	$Ft_i : \{graph \mid object \mid noVertex \mid domain \mid lenght\}$
Threshold T_h	Specification	Use Case	Analyzer	$T_{hi} : T_h \quad name \mid object$
A. Threshold At_h	Specification	Use Case	Analyzer	$AT_{hi} : AT_h \quad name \mid data \quad structure \mid CI \mid forecast$
Datasets D	Specification	Use Case	Analyzer	$D_i : \{D \quad name \mid object \mid type \mid source\}$
Pattern Recognition	Specification	Use Case	Analyzer	$PR_i : \{PR \quad name \mid objectIn \mid ObjectOut \mid action \mid reference \quad data\}$
Conclusion C	Specification	Use Case	Analyzer	$C_i : \{C \quad name \mid use \quad case \mid fact\}$
Report Re	Report	Analyzer	Diagnosis	$Re_i : \{C \quad name \mid use \quad case \mid fact \mid uncertainty \mid trigger\}$

5.1. Object O

The objects $O = \{O_1, ..., O_n\}$, $n \geq 1$ are definitions of the elements from which the system infers knowledge. They are added to the knowledge-base by use case operators. Their function is to describe the nature of the data in order to facilitate the selection of proper preprocessing and prediction methods. Objects are expressed as follows:

$$O_i : \{object \quad name \mid weight \mid noValues \mid range \quad of \quad values \quad Va\} \tag{1}$$

Examples:
$\{Temperature \mid 1 \mid 1 \mid (-30°, 150°)\}$,
$\{Link \quad Status \mid 0.7 \mid 1 \mid \{"Good", "Normal", "Bad"\}\}$,
$\{Header \quad Encryption \mid 1.5 \mid 1 \mid (True, False)\}$,
$\{Upper \quad Threshold \mid 2 \mid 1 \mid Y_t : t \epsilon T, \forall Y_i \epsilon R\}$

where *object name* acts as identification of the data category and the range of values limits the values that can be assigned. The *weight* is a field reserved for the future implementation of machine learning; it determines priority. Finally, *noValues* anticipates its amount of possible values. Because of this, an object may be specified as a sequence of k previously defined objects or values interrelated. In this case, they are defined as follows.

$$O_i : \{object \quad name \mid weight \mid noValues \mid [Va_1][Va_2]...[Va_k]\} \tag{2}$$

Examples:
$\{pairWeather \mid 1 \mid 2 \mid [temperature][humidity]\}$,
$\{metricA \mid 2 \mid 4 \mid [TTL][lenght][port][ipAddress]\}$,
$\{tSerieB \mid 2 \mid 8 \mid [R][R][R][R][R][R][R][R]\}$,

The specification of sequences of the same value repeated several times in a row can be simplified by the indicator : i, where i is the number of times it repeats. For example, the previous example:

$$\{tSerieB \mid 2 \mid 8 \mid [R][R][R][R][R][R][R][R]\},$$

It may be simplified as follows:

$$\{tSerieB \mid 2 \mid 8 \mid [R] : 8\},$$

5.2. Operations Op

The operations $Op = \{Op_1, ..., Op_n\}$, $n \geq 1$ are definitions of binary relationships between facts Fa, objects O or their possible values Va. Initially, the knowledge-base provides a basic battery of operations (ex. All arithmetic operations, propositional logic relationships, basic statistic expressions, etc.). When a use case is on-boarded, operators should declare the set of operations to be taken into account and their restrictions. This is achieved by the following layout:

$$Op_i : \{name \mid symbol \mid priority \mid operands \mid description\} \tag{3}$$

Examples:
$\{Equal \mid = \mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$,
$\{LGT \mid \geq \mid 1 \mid (Fa, O, Va) \geq (Fa, O, Va) \mid left \quad is \quad GE\}$,
$\{And \mid \wedge \mid 1 \mid (Fa) \wedge (Fa) \mid logical \quad conjuction\}$,
$\{Addition \mid + \mid 3 \mid (Fa, Vo) + (Fa, Vo) \mid addition\}$,

where *name* refer to the identification of the operation in the predefined battery, *symbol* is its shortened representation, *priority* its position in the hierarchy of operations, *operands* limits the categories

of operands applicable on each side of the binary expression, and *description* briefly explains its functionality in natural language.

5.3. Facts Fa

The facts $Fa = \{Fa_1, ..., Fa_n\}$, $n \geq 1$ are the basic elements of the SELFNET reasoning. They are added to the memory of the Analyzer Module by the Aggregation layer or deduced by the inference engine. Facts are constructed by the linear grammar $GFa = (N, \sum, P, State)$ where $N = State, Operand, \sum = O, Op, Va$ and P is extended as:

$$State \longrightarrow A \quad op \quad A$$
$$Operand \longrightarrow object \mid fact \mid value$$

Facts must be accompanied by a *timestamp* indicating when they have been stated, the *location* on which they are valid and a *weight* that determine their priority. The location refers to SELFNET elements (ex. physical machines, virtual nodes, etc.). The priority is a field reserved by future machine learning weighting. *Uncertainty* describes its probability of being true. Facts are described as the following expression:

$$Fa_i : \{expresion \mid weight \mid uncertainty \mid timestamp \mid location\} \tag{4}$$

Examples:
$\{Ur_{threshold} = MaxValue \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid NodeA\}$,
$\{Temperature \geq 80° \mid 0.7 \mid 0.98 \mid Today \quad 03:41:20 \mid VM15\}$,
$\{KPI7 = UrTh + MaxT \mid 1.2 \mid 1 \mid Today \quad 03:41:20 \mid VM15\}$,

5.4. Rules Ru

The $rules = \{Ru_1, ..., Ru_n\}, n \geq 1$ describe how the Analyzer Module acquires new knowledge via rule-based expert system. In order to facilitate their specification, they are declared as propositional logic expressions, and according with the linear grammar $GRu = (N, \sum, P, Rule)$, where $\sum = "True", "False", Facts, N = Rule, Atomic \mid Symbol \mid Complex$, and P is expressed as follows:

$$Rule \longrightarrow Atomic \mid Complex \tag{5}$$

$$Atomic \longrightarrow "True" \mid "False" \mid Symbol$$
$$Symbol \longrightarrow Facts$$
$$Complex \longrightarrow \neg Rule \mid (Rule \longrightarrow Rule) \mid (Rule \leftrightarrow Rule) \mid (Rule \wedge Rule) \mid (Rule \vee Rule)$$

Examples:
$(Fa(X) \vee Fa(Y)) \longrightarrow Fa(Z)$
$(Fa(X) \wedge Fa(Y)) \vee (Fa(X) \wedge \neg Fa(Z)) \longrightarrow Fa(B)$
$(Fa(C) \vee Fa(Y)) \vee \neg(Fa(A) \wedge Fa(Z)) \longrightarrow Fa(B)$

The rules are accompanied by the identification of the *usecase* on which they are valid, and their *priority* of inference. Note that in order to enhance scalability, the rules of each use case are totally independent from the others. Rules are detailed as follows:

$$Rule : \{rule \mid priority \mid use \quad case\} \tag{6}$$

Examples:
$\{(Fa(X) \vee Fa(Y)) \longrightarrow Fa(Z) \mid 1 \mid SP\}$
$\{(Fa(B)) \longrightarrow Fa(Y) \mid 2 \mid SO\}$
$\{(Fa(X) \wedge Fa(Y)) \vee (Fa(X) \wedge \neg Fa(Z)) \longrightarrow Fa(B) \mid 1 \mid SH\}$

5.5. Forecast Ft

The $Forecasts = \{Ft_1, ..., Ft_n\}$, $n \geq 1$ are specifications of the objects that must be projected per use case. In this way it is possible to enhance the selection of prediction algorithms and forecasting models. Given the nature of the monitoring environment, *a priori*, this approach only considers predictions on two data types: time series and graphs. The time series allow estimating the evolution of Key Performance Indicators (KPI) or thresholds from concrete locations on SELFNET (physical infrastructure, network devices, virtualization, etc.). The prediction on graphs facilitates the inference of changes on large regions of the SELFNET topology, such as spreading of congestion, inclusion of new network elements or failures. This expert system considers prediction results as facts, so Ft only refers to their specification when on-boarding new use cases. In Figure 1 predictions as facts are declared as $Fa(Ft)$. The following expression describes the forecasts on time series:

$$Ft_i : \{timeSeries \mid object \mid domain \quad lenght\} \tag{7}$$

Examples:
$\{timeSeries \mid O_1 \mid obs \mid t + 5\}$
$\{timeSeries|O_2 \mid time \mid Today \quad 13:28:15\}$

where *timeSeries* is a reserved word indicating that the prediction is on time series, *object* declares the nature of the data to be analyzed, and *domain* is the extension of the prediction. The examples show two reserved words, *obs* (observations) and *time* (timestamp). When the time is measured in observations, the length of the prediction is indicated from the initial time instant t and the amount of coming observations (ex. $t + 5$ indicates forecast the next five observations). On the other hand, timestamps directly detail how long must be the prediction (ex. Today 13:28:15 indicates the requirement of forecast a certain object between now and 13:28:15 today). Note that the term *timeSeries* is used to describe the way in which data is structured and not the prediction algorithm. A record tracking of this nature could be forecasted by traditional time series methods (autoregressive moving average, exponential smoothing, extrapolation, etc.) but also by other very different approaches (drifting, naive-based algorithms, Artificial Neural Networks-ANN, Support Vector Machines-SVM, etc.). It is up to the decision component of Prediction, select the most appropriate forecasting strategy. If the prediction considers observations on graphs, the forecasts are specified as follows:

$$Ft_i : \{graph \mid object \mid noVertex \mid domain \mid lenght\} \tag{8}$$

Examples:
$\{graph \mid O_1 \mid 30 \mid obs \mid t + 20\}$
$\{graph \mid O_2 \mid 45 \mid timestamp \mid Today \quad 19:12:07\}$
$\{graph \mid O_3 \mid 10 \mid timestamp \mid Today \quad 22:30:00\}$

where *graph* is the reserved word to declare predictions on graphs. *object* is the nature of the data on the edges of its incidence matrix. *noVertex* is the number of vertex (i.e., dimension *noVertex-by-noVertex* of its complete adjacency matrix). The last two parameters (*domain* and *length*) have the same function as in the expression of *timeSeries* prediction (indicate the measurement of time and the extension of the prediction).

5.6. Thresholds T_h

The thresholds $T_h = \{T_{h1}, ..., T_{hn}\}$, $n \geq 1$ are specifications of fault tolerance limits related with values assigned to objects O. They are calculated by the SELFNET Aggregation task, but their specification is part of the use case operators. Thresholds are described as the following expression:

$$T_{hi} : T_h \quad name \mid object \tag{9}$$

Examples:
$\{maxTemp \mid O(temperature)\}$
$\{maxConnections \mid O(nConnections)\}$
$\{minQuality \mid O(QoS)\}$

where T_h name is the threshold identification and *object* is the object on which it acts.

5.7. Adaptive Thresholds T_h

The adaptive thresholds $AT_h = \{AT_{h1}, ..., AT_{hn}\}$, $n \geq 1$ are specification of fault tolerance limits related with values assigned to predictions Ft. They are calculated by the component of prediction of the Analyzer Module, but must be specified by the use case operators. Similarly to the forecast descriptions, initially they act on time series or graphs. They are described as follows:

$$AT_{hi} : AT_h \quad name \mid data \quad structure \mid CI \mid forecast \tag{10}$$

Examples:
$\{maxTemp \mid timeSeries \mid 0.95 \mid Ft(A)\}$
$\{maxWorkload \mid graph \mid 0.90 \mid Ft(X)\}$

where $AT_h \quad name$ is the identification of the adaptive threshold, $data \quad structure$ is *timeSeries* or *graph* depending on the representation of the predicted data, CI is the confidence interval on which it is built by the Adaptive Thresholding component and forecast is the prediction from which it is created.

5.8. Pattern Recognition PR

The pattern recognition configurations $PR = \{PR_1, \ldots, PR_n\}$, $n \geq 1$ are specifications of how facts Fa related with aggregate data are analyzed in order to determine their similarity with previously established reference information. The outputs of pattern recognition actions are facts that display the degree of the similarity observed. Each PR action is defined as follows:

$$PR_i : \{PR \quad name \mid objectIn \mid ObjectOut \mid action \mid reference \quad data\} \tag{11}$$

Examples:
$\{botnetTraffic \mid O(tFlow) \mid O(dist) \mid match \mid D(dataset1)\}$
$\{paylScan \mid O(payload) \mid O(dist) \mid anomaly \mid D(dataset2)\}$
$\{usrVerify \mid O(uAction) \mid O(dist) \mid anomaly \mid D(dataset3)\}$

where $PR \quad name$ is the action identificator, *objectIn* is the nature of the data to be studied, *objectOut* is the nature of the object recipient of the similarity degree, *action* is the reserved word associated with the type of analysis to be performed. The default actions are "match" for matching observations with the reference data and "anomaly" for outlier detection. Finally, $referencedata$ is the identification of the dataset D to be taken into account.

5.9. Datasets D

The Datasets $D = \{D_1, \ldots, D_n\}$, $n \geq 1$ is the initial reference data to be required by pattern recognition actions. Given that Analyzer Module does not consider online training, all the reference data is provided by the use cases via User Interface. Datasets are declared by the following expression:

$$D_i : \{D \quad name \mid object \mid type \mid source\} \tag{12}$$

Examples:
$\{legitimatePayload \mid O(payload) \mid model \mid Repository1\}$
$\{mySet1 \mid O(flowMetrics) \mid collection \mid Repository2\}$
$\{autoreplicationGens \mid O(binary) \mid signature \mid Repository3\}$

where $D \quad name$ is the dataset identifier and *object* is the nature of its samples. In this first approach, the dataset can be framed by three types: "collection", "model" or "signature". Firstly, "collection" refers to a set of raw observations directly extracted from the monitoring environment. On the other hand, "model" is a preprocessed description of the data to be analysed. Finally, "signature" indicates exactly patters to be identified. The field *source* determines where the dataset is found (ex. path, url, repository, etc.).

5.10. Conclusions C

The conclusions $C = \{C_1, \ldots, C_n\}$, $n \geq 1$ are the subset of the group of facts Fa specified for a use case to be satisfied, that form part of the Situational Awareness of the network. When a conclusion is inferred, it is reported to the Diagnostic module [60] for being a potential indicator of situations. These symptoms are defined by use case operators as follows:

$$C_i : \{C \quad name \mid use \quad case \mid fact\} \tag{13}$$

Examples:
$\{gridlock \mid SP \mid Fa(A)\}$
$\{overHeating \mid SH \mid fa(X)\}$

where C name is the conclusion identificator, use case is the associated SELFNET use case, and fact is the triggering conclusion. Conclusions are reported to Diagnostic Module as follows:

$$Re_i : \{C \quad name \mid use \quad case \mid fact \mid uncertainty \mid trigger\} \tag{14}$$

Examples:
$\{gridlock \mid SP \mid Fa(A) \mid 0.85 \mid Fa(B), Fa(C), Ru(1)\}$
$\{overHeating \mid SH \mid fa(X) \mid 0.75 \mid Fa(x), Ru(3)\}$

where *uncertainty* the probability of being certain and trigger is the list of rules Ru or facts Fa that take part of its inference.

6. Examples of Specification and Workflows

This section describes three examples of data specification and workflows on the Analyzer Module.

6.1. UC 1: Device Temperature Analysis

This section describes an example of a sensor related with self-healing use case.

6.1.1. Description

The use case (*myTemp*) requires identifying symptoms related with overheat on network devices. This is a very basic example where prediction and adaptive thresholding are not considered. Therefore the decision thresholds are static and were built at Aggregation.

6.1.2. Initial Status

The Analyzer Module disposes of a battery of predefined operations, including basic arithmetic calculations, logical and statistical functions.

6.1.3. Use Case Specification

First, the use case operators specify the basic objects to be taken into account: the temperature of the devices and its upper threshold.

$$O_1 : \{Temperature \mid 1 \mid 1 \mid R\}$$
$$T_{h1} : \{maxTemp \mid O_1\}$$

Second is indicating the operators that are required and how they are taken into account:

$$Op_1 : \{Equal \mid = \mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$$
$$Op_2 : \{LGT \mid \geq \mid 1 \mid (Fa, O, Va) \geq (Fa, O, Va) \mid leftisGE\}$$

Third, the conclusions to be satisfied:

$$C_1 : \{overheat \mid myTemp \mid Fa(O_1) \geq Fa(T_{h1})\}$$

The last step is declaring the inference rules:

$$Ru_1 : \{Fa(O_1) \geq Fa(T_{h1}) \longrightarrow Fa(C_1) \mid 1 \mid myTemp\}$$

6.1.4. Workflow

At runtime, Aggregation layer notify to the Analyzer Module facts related with myTemp use case. Some of them concern the temperature on SELFNET devices, for example:

$$Fa_1 : \{O_1 = 35^\circ \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid NodeA\}$$
$$Fa_2 : \{O_1 = 76^\circ \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid NodeB\}$$
$$\dots$$
$$Fa_5 : \{O_1 = 80^\circ \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid NodeB\}$$

Note that uncertainty is 1 because the sensors are deterministic (100% probability of provide the correct temperature). The facts refer to the static thresholding are:

$$Fa_7 : \{T_{h1} = 79^\circ \mid 1 \mid 1 \mid Today \quad 12:22:15 \mid All\}$$
$$Fa_8 : \{T_{h1} = 79^\circ \mid 1 \mid 1 \mid Today \quad 12:22:16 \mid All\}$$

These facts are provided by Aggregation, and they are directly included on the memory of the Analyzer Module. If they are updated for the same location (ex. Fa_5 and Fa_6), the latest version is considered by the inference engine. After certain period of observation, the inference engine tries to deduct new knowledge from the rule-set of every use case. In myTemp, the Analyzer Module tries to infer conclusions for Ru_1. At $Today \quad 12:22:17$ the systems satisfy the first conclusion: $Fa_5(O_1 = 80^\circ) \geq Fa_8(O_1 = 79^\circ)$, so the fact $Fa(C_1)$ is added to memory:

$$Fa_9 : \{Fa_5 \geq Fa_8 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid NodeB\}$$

The location NodeB is considered because it is the more restrictive between NodeB,All. So the symptom C_1 has been discovered, and it is reported to Diagnostic Module as follows:

$$Re_1 : \{overheat \mid myTemp \mid Fa_9 \mid 1 \mid Fa_5, Fa_8, Ru_1\}$$

The inference engine will continue operating looking for new symptoms.

6.2. UC 2: Network Congestion Analysis

This section describes an example of a sensor related with self-optimization use case.

6.2.1. Description

The use case to be managed (Self-Congestion (SC)) requires identifying symptoms related with traffic congestion on SELFNET elements. In this example, prediction and adaptive thresholding are considered.

6.2.2. Initial Status

The Analyzer Module disposes of a battery of predefined operations, including basic arithmetic calculations, logical and statistical functions.

6.2.3. Use Case Specification

First, the use case operators specify the basic objects to be taken into account: the congestion level monitored and its prediction.

$$O_1 : \{congestion \mid 1 \mid 1 \mid [0,1]\}$$
$$Ft_1 : \{timeSeries \mid O_1 \mid obs \mid t+3\}$$

Next, they define an adaptive threshold to be automatically generated from the information provided by the record tracking and the Adaptive Thresholding.

$$AT_{h1} : \{maxCongestion \mid timeSeries \mid 0.95 \mid Ft_1\}$$

Second, it is specified what operators are required and how they are taken into account:

$$Op_1 : \{Equal \mid = \mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$$
$$Op_2 : \{LGT \mid \geq \mid 1 \mid (Fa, O, Va) \geq (Fa, O, Va) \mid leftisGE\}$$

Third, the conclusions are identified:

$$C_1 : \{gridlock \mid SC \mid Fa(O_1) \geq Fa(AT_{h1})\}$$

The last step is declaring the inference rules:

$$Ru_1 : \{Fa(O_1) \geq Fa(AT_{h1}) \longrightarrow Fa(C_1) \mid 1 \mid SC\}$$

6.2.4. Workflow

At runtime, Aggregation layer notify to the Analyzer Module facts related with the SC use case, for example:

$$\begin{aligned}
Fa_1 &: \{O_1 = 0.6 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ServerA\} \\
Fa_2 &: \{O_1 = 0.65 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ServerA\} \\
Fa_3 &: \{O_1 = 0.65 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ServerA\} \\
Fa_4 &: \{O_1 = 0.67 \mid 1 \mid 1 \mid Today \quad 12:22:21 \mid ServerA\} \\
Fa_5 &: \{O_1 = 0.68 \mid 1 \mid 1 \mid Today \quad 12:22:24 \mid ServerA\} \\
&\ldots. \\
Fa_{44} &: \{O_1 = 0.66 \mid 1 \mid 1 \mid Today \quad 12:22:50 \mid ServerA\} \\
Fa_{45} &: \{O_1 = 0.67 \mid 1 \mid 1 \mid Today \quad 12:22:52 \mid ServerA\} \\
Fa_{47} &: \{O_1 = 0.69 \mid 1 \mid 1 \mid Today \quad 12:22:56 \mid ServerA\} \\
Fa_{48} &: \{O_1 = 0.86 \mid 1 \mid 1 \mid Today \quad 12:22:58 \mid ServerA\} \\
Fa_{49} &: \{O_1 = 0.97 \mid 1 \mid 1 \mid Today \quad 12:23:01 \mid ServerA\}
\end{aligned}$$

The construction of predictive models requires certain amount of previous observations; in this case, it considered the first 45 facts. They are handled by the record tracking in order to extract the needed information and define time series. At *Today* 12 : 22 : 52 (when Fa_{45} is deducted), the forecasting component provides the first prediction Ft_1 for the instant t+3. Then a new fact is included to the memory:

$$Fa_{46} : \{AT_{h1} = 90 \mid 1 \mid 1 \mid Today \;\; 12:23:52 \mid ServerA\}$$

It triggers the rule Ru_1, because $Fa_49(O_1 = 0.97) \geq Fa_46(AT_{h1} = 90)$, and the conclusion C is satisfied. The new knowledge $Fa(C_1)$ is added to memory as:

$$Fa_{50} : \{Fa_{49} \geq Fa_{46} \mid 1 \mid 1 \mid Today \;\; 12:23:01 \mid NodeB\}$$

Finally, the symptom is reported to Diagnostic Module as follows:

$$Re_1 : \{gridlock \mid SC \mid Fa_{50} \mid 1 \mid Fa_{49}, Fa_{46}, Ru_1\}$$

6.3. UC 3: Payload Analysis

This section describes an example of a sensor related with self-protection use case.

6.3.1. Description

This use case (Self-Guard(SG)) requires identifying symptoms related with anomalous payloads on SELFNET traffic. In this example, pattern recognition actions are considered.

6.3.2. Initial Status

The Analyzer Module disposes of a battery of predefined operations, including basic arithmetic calculations, logical and statistical functions. The external repositories (Rep1, Rep2) provide collection of Legitimate (Rep1) and malicious (Rep2) SELFNET traffic observations.

6.3.3. Use Case Specification

First, the use case operators specify the basic objects to be taken into account; in this example they are the payload of the SELFNET traffic O_1, its similarity with the legitimate payload dataset O_2 and the malicious samples O_3.

$$O_1 : \{payload \mid 1 \mid 1 \mid hexadecimal\}$$
$$O_2 : \{simLegi \mid 1 \mid 1 \mid \{0...1\}\}$$
$$O_3 : \{simMal \mid 1 \mid 1 \mid \{0...1\}\}$$

Second, it is specified what operators are required and how they are taken into account:

$$Op_1 : \{Equal \mid = \mid 1 \mid (Fa, O, Va) = (Fa, O, Va) \mid equal\}$$
$$Op_2 : \{LT \mid > \mid 1 \mid (Fa, O, Va) > (Fa, O, Va) \mid leftisG\}$$

Next, they define the datasets to be taken into account.

$$D_{legi} : \{legitimatePayload \mid O(payload) \mid collection \mid Rep1\}$$
$$D_{mal} : \{maliciousPayload \mid O(payload) \mid collection \mid Rep2\}$$

And then the pattern recognition actions to be executed:

$$PR_1 : \{legMeasure \mid O_1 \mid O_2 \mid anomaly \mid D(D_{legi})\}$$
$$PR_2 : \{malMeasure \mid O_1 \mid O_3 \mid anomaly \mid D(D_{mal})\}$$

Conclusions are identified as follows:

$$C_1 : \{maliciousContent \mid SC \mid Fa(O_2) < Fa(O_3)\}$$

And the following rules are onboarded:

$$Ru_1 : \{Fa(O_2) < Fa(O_3) \longrightarrow Fa(C_1) \mid 1 \mid SP\}$$

6.3.4. Workflow

At runtime, Aggregation layer notifies to the Analyzer Module facts related with the SG use case:

$$Fa_1 : \{O_1 = FF217 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionA\}$$
$$Fa_2 : \{O_1 = FFFFF \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionB\}$$
$$Fa_3 : \{O_1 = 00DE8 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionA\}$$
$$Fa_4 : \{O_1 = A4FC9 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionA\}$$
$$Fa_5 : \{O_1 = FF218 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ConexionC\}$$
$$\ldots.$$
$$Fa_{38} : O_1 = F0279 \mid 1 \mid 1 \mid Today \quad 12:22:23 \mid ConexionA$$

Each time a new payload is observed, the SELFNET Analyzer Module performs the pattern recognition actions PR_1 and PR_2. This returns new facts:

$$\begin{array}{c} Fa_{1R1} : \{O_2 = 0.9 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionA\} \\ Fa_{1R2} : \{O_3 = 0.2 \mid 1 \mid 1 \mid Today \quad 12:22:17 \mid ConexionA\} \\ Fa_{2R1} : \{O_2 = 0.9 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionB\} \\ Fa_{2R2} : \{O_3 = 0.18 \mid 1 \mid 1 \mid Today \quad 12:22:18 \mid ConexionB\} \\ Fa_{3R1} : \{O_2 = 0.8 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ConexionC\} \\ Fa_{3R2} : \{O_3 = 0.21 \mid 1 \mid 1 \mid Today \quad 12:22:19 \mid ConexionC\} \\ \ldots . \end{array}$$

At *Today* 12 : 22 : 23 the following facts are discovered:

$$\begin{array}{c} Fa_{32R1} : \{O_2 = 0.66 \mid 1 \mid 1 \mid Today \quad 12:22:23 \mid ConexionA\} \\ Fa_{32R2} : \{O_3 = 0.92 \mid 1 \mid 1 \mid Today \quad 12:22:23 \mid ConexionA\} \end{array}$$

They trigger the rule Ru_1, because $Fa_{32R1}(O_2 = 0.66) < Fa_{32R2}(O_3 = 0.92)$, and then the conclusion C is satisfied. The new knowledge $Fa(C_1)$ is added to memory as:

$$Fa_{50} : \{Fa_{32R1} < Fa_{32R1} \mid 1 \mid 1 \mid Today \quad 12:23:23 \mid ConexionA\}$$

Finally, the symptom is reported to Diagnostic Module as follows:

$$Re_1 : \{suspiciousPayload \mid SC \mid Fa_{50} \mid 1 \mid Fa_{32R1}, Fa_{32R2}, Ru_1\}$$

7. Discussion

Analysis and intelligence capabilities play an important role to address 5G requirements, in combination with key-enabled technologies such as SDN, NFV, cloud computing, etc. All of these domains can take advantage of forecasting, pattern recognition, artificial intelligence and advanced intelligence concepts. In this way, 5G networks will be able to provide enhanced capacities related to the network management and the detection of possible harmful problems. For its part, the diagnosis of data information is required in order to know what the real cause of the event is. Because of this, the intelligence is provided in two phases: (i) analysis stage and (ii) decision-making; similar to a medical evaluation, where firstly the symptoms are detected and then based on it a treatment is applied. In general terms, the proposed architecture and data specification for enhancing a use-case driven analysis on 5G networks presupposes substantial improvements over previous approaches. On the one hand, 5G data analysis is partially covered in some works [27,30,31]. These proposals take into account specific requirements such as context awareness of radio components [27]. In [30] a context aware resource allocation algorithm based on the user mobility is presented. This work proposes some resource management schemes, handover procedures and cell activations. For its part, Apajalahti et al. [31] use ontologies and statistical reasoning in order to analyze and configure the mobile network. This approach can be used as a complementary methodology to the SELFNET Analyzer approach. On the other hand, some ongoing works [28,29,36] are still at an early stage and are complementary to our proposal. METIS project [28] is dealing with 5G radio access network components (e.g., spectrum usage or air interface). In [29] intelligent capabilities applied to virtualized environments are proposed. Furthermore, Charisma Project [36] introduces intelligent mobile cloud in order to meet low latency and security requirements.

To the best of our knowledge, the SELFNET Analyzer framework is the first proposal that provides a generalized framework to deal with both traditional technologies and currently 5G key-enabled technologies. The SELFNET Analyzer Module provides a general purpose scheme easily adapted to the operator needs and hence to overcome the design constraints in different monitoring environments. This is a very important feature bearing in mind the great amount of technologies that can be part of a 5G scenario, as well those that still under development to be deployed in the near future [10,14,15].

Note that SELFNET Analyzer Module is able to analyse information from heterogeneous sources such as SDN elements, virtual devices or metrics from specialized sensors. Another important characteristic is that the SELFNET Analyzer facilitates the incorporation of different analysis strategies, such as novel prediction or pattern recognition algorithms. This framework was developed to be able to operate indistinctly with very different data mining and machine learning paradigms, among them conventional information, big data or high dimensional data. The use of any of them does not imply design changes, being simply an implementation problem. As a result, this proposal is easily adaptable to future projects.

The proposed data specification to accommodate the onboarding of new use cases is simple and adjustable. This is also corroborated by the fact that SELFNET has been able to incorporate every use case without modifications on the original definitions. This does not mean that in future use cases, more relevant changes will not be required. But without doubt, this robustness provides a solid base for design analytic schemes on similar contexts. Note that SELFNET implements a triad of services: self-protection, self-healing and self-optimization with completely different features and dependences (metrics, network devices to be monitored, prediction/pattern recognition algorithms, etc.). But despite these advantages, the proposal presents some weakness, most of them related with the limitations previously mentioned at the design principles (see Section 3). For example, the SELFNET Analyzer is not able to deal with complex stationary monitoring environments [42], where the quality of the analytics will decrease with time. Given the importance of this kind of scenarios on network environments, this is an aspect that must be studied.

Another point to be keep in mind is that, according to the experience of the SELFNET consortium, the effectiveness of the Analyzer Module depends on the quality of their specification. It means that once deployed, the approach follows the guidelines provided by operators, which indicate what information should be processed, how it should be analysed and what results can be obtained from it. Despite of the simplicity of the proposed data specification, if the operator makes errors, there is a greatest chance of unexpected results. So in this sense, its robustness and scalability imply a high dependence on the quality of the specification inserted by operators where configuring the analyzer functionality. It is important to emphasize that loading new use cases is based only on configuration changes, without the need to modify the Analyzer implementation or to include additional software. Furthermore, there are a number of challenges that need to be addressed related to how the data received from underlying layers will be organized or how the analysis process will be performed. Regarding to data organization is important to determine if the data will be processed as a raw data or in an aggregated manner because it may become a performance issue. This information will be loaded and converted into facts by the Analyzer framework in order to provide the network state in real time. Another important aspect to bear in mind is the execution pipeline of the Analyzer components in order to provide consistence and facilitate the organization of the received information. Thus, the investigation of methods to process and analyze the received information is also part of the ongoing work.

8. Conclusions

In this paper, the application of analysis and intelligence capabilities and how these concepts are used in 5G networks were explained. We introduced the design of SELFNET Analyzer Module and its data specification. Our design provides pattern recognition, reasoning and prediction capabilities to infer the possible symptoms, facilitating the diagnosis and decision-making tasks, which are part of future work. The main contribution of SELFNET Analyzer Module is its general, simple and scalable approach, allowing new rules and metrics in the analysis process when a new use case is added by the operator. SELFNET sensors gather information from several data sources such as virtual elements, LTE, SDN and traditional network devices; and thus the gathered information can be subject of analysis. Furthermore, this proposal was built to support new analytic capabilities by means of a plugin based

approach. Meanwhile, the implementation of Analyzer Module is part of ongoing work as well as the introduction of mechanisms to work in non-stationary monitoring environments.

Acknowledgments: This work is supported by the European Commission Horizon 2020 Programme under grant agreement number H2020-ICT-2014-2/671672-SELFNET (Framework for Self-Organized Network Management in Virtualized and Software Defined Networks). Lorena Isabel Barona López are supported by the Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación SENESCYT (Quito, Ecuador) under Convocatoria Abierta 2013 Scholarship Program.

Author Contributions: The authors contributed equally to this research. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. NGMN Alliance. NMGN 5G White Paper 2015. Available online: https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0.pdf (accessed on 9 January 2017).
2. Agyapong, P.K.; Iwamura, M.; Staehle, D.; Kiess, W.; Benjebbour, A. Design Considerations for a 5G Network Architecture. *IEEE Commun. Mag.* **2014**, *52*, 65–75.
3. Kreutz, D.; Ramos, F.M.V.; Verissimo, P.E.; Rothenberg, C.E.; Azodolmolky, S.; Uhlig, S. Software-defined Networking: A Comprehensive Survey. *Proc. IEEE* **2015**, *103*, 14–76.
4. Hu, F.; Hao, Q.; Bao, K. A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1–27.
5. ETSI Industry Specification Group (ISG). Network Function Virtualization (NFV) Architectural Framework. Available online: http://www.etsi.org/technologies-clusters/technologies/nfv (accessed on 9 January 2017)
6. Mijumbi, R.; Serrat, J.; Gorricho, J.L.; Bouten, N.; Turck, F.; Boutaba, R. Network Function Virtualization: State-of-the-art and Research Challenges. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 236–262.
7. Zhang, Q.; Cheng, L.; Boutaba, R. Cloud Computing: State-of-the-art and Research Challenges. *J. Int. Serv. Appl.* **2010**, *1*, 7–18.
8. Baldo, N.; Giupponi, L.; Mangues-Bafalluy, J. Big Data Empowered Self Organized Networks. In Proceedings of the 20th European Wireless Conference, Barcelona, Spain, 14–16 May 2014.
9. Aliu, O.G.; Imran, A.; Imran, M.A.; and Evans, B. A Survey of Self Organisation in Future Cellular Networks. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 336–361.
10. Imran, A.; Zoha, A.; Abu-Dayya, A. Challenges in 5G: How to Empower SON with Big Data for enabling 5G. *IEEE Netw.* **2014**, *28*, 27–33.
11. Quick, D.; Choo, K.K.R. Digital forensic Intelligence: Data Subsets and Open Source Intelligence (DFINT + OSINT): A timely and Cohesive Mix. *Future Gener. Comput. Syst.* **2016**, doi:10.1016/j.future.2016.12.032.
12. Quick, D.; Choo, K.K.R. Big Forensic Data Management in Heterogeneous Distributed Systems: Quick Analysis of Multimedia Forensic Data. Software: Practice and Experience. *J. Netw. Comput. Appl.* **2016**, doi:10.1002/spe.2429.
13. Demestichas, P.; Georgakopoulos, A.; Karvounas, D.; Tsagkaris, K.; Stavroulaki, V. 5G on the horizon: Key Challenges for the Radio-Access Network. *IEEE Veh. Technol. Mag.* **2013**, *8*, 47–53.
14. Boccardi, F.; Heath, R.W.; Lozano, A.; Marzetta, T.L.; Popovski, P. Five Disruptive Technology directions for 5G. *IEEE Commun. Mag.* **2014**, *52*, 74–80.
15. Akyildiz, I.F.; Lin, S.C.; Wang, P. Wireless Software-Defined Networks (W-SDNs) and Network Function Virtualization (NFV) for 5G Cellular Systems: An Overview and Qualitative Evaluation. *Comput. Netw.* **2015**, *93*, 66–79.
16. Abdelwahab, S.; Hamdaoui, B.; Guizani, M.; Znati, T. Network Function Virtualization in 5G. *IEEE Commun. Mag.* **2016**, *54*, 84–91.
17. Lin, X.; Choo, K.K.R.; Lin, Y.D.; Mueller, P. Guest Editorial: Network Forensics and Surveillance for Emerging Networks. *IEEE Netw.* **2016**, *30*, 4–5.
18. 5G Infrastructure Public Private Partnership—5G PPP. Available online: https://5g-ppp.eu (accessed on 16 December 2016).
19. 5G Americas, 2016. Available online: http://www.5gamericas.org/es/ (accessed on 16 December 2016).

20. Barona López, L.I.; Valdivieso Caraguay, Á.L.; Maestre Vidal, J.; Sotelo Monge, M.A.; García Villalba, L.J. Towards Incidence Management in 5G based on Situational Awareness. *Future Internet* **2017**, *9*, 1–15.
21. 5G Ensure. Deliverable D 2.3, Risk Assessment, Mitigation and Requirements (Draft). Available online: http://www.5gensure.eu/deliverables (accessed on 19 December 2016).
22. SELFNET Project. Framework for Self-Organized Network Management in Virtualized and Software Defined Networks. Available online: https://selfnet-5g.eu/ (accessed on 15 February 2017).
23. Tahaei, H.; Salleh, R.; Khan, S.; Izard, R.; Choo, K.K.R.; Anuar, N.B. A multi-objective Software Defined Network Traffic Measurement. *Measurement* **2016**, *95*, 317–327.
24. Agiwal, M.; Roy, A.; Saxena, N. Next Generation 5G Wireless Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1617–1655.
25. Liu, D.; Wang, L.; Chen, Y.; Elkashlan, M.; Wong, K.K.; Schober, R. User association in 5G networks: A survey and an outlook. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1018–1044.
26. Fenton, N.; Neil, M. Making decisions: Using Bayesian Nets and MCDA. *Knowl. Syst.* **2001**, *14*, 307–325.
27. Marquezan, C.C.; Mahmood, K.; Zafeiropoulos, A.; Krishna, R.; Huang, X.; An, X.; Corujo, D. Context Awareness in Next Generation of Mobile Core Networks. Available online: https://arxiv.org/ftp/arxiv/papers/1611/1611.05353.pdf (accessed on 9 January 2017).
28. Tullberg, H.; Popovski, P.; Li, Z.; Uusitalo, M.A. The METIS 5G System Concept–Meeting the 5G Requirements. *IEEE Commun. Mag.* **2016**, *54*, 132–139.
29. CONTENT Project. Convergence of Wireless Optical Network and iT rEsources iN SupporT of Cloud Services. FP7-ICT. Project Reference: 318514, Funded under: FP7-ICT. Available online: http://cordis.europa.eu/fp7/ict/future-networks/documents/call8-projects/content-factsheet.pdf (accessed on 9 January 2017).
30. Kuruvatti, N.P.; Schotten, H.D. Framework to Support Mobility Context Awareness in Cellular Networks. In Proceedings of the IEEE 83rd Vehicular Technology Conference (VTC Spring), Nanjing, China, 15–18 May 2016.
31. Apajalahti, K.; Eero, H.; Juha, N.; Vilho, R. StaRe: Statistical Reasoning Tool for 5G Network Management. In Proceedings of the 2016 Semantic Web-ESWC, Heraklion, Greece, 29 May–2 June 2016.
32. Martin, B.A.; Marinos, L.; Rekleitis, E.; Spanoudakis, G.; Petroulakis, N.E; Threat Landscape and Good Practice Guide for Software Defined Networks/5G. Available online: http://openaccess.city.ac.uk/15504/7/SDN%20Threat%20Landscape.pdf (accessed on 9 January 2017).
33. You, I.; Sharma, V.; Atiquzzaman, M.; Choo, K.K.R. GDTN: Genome-Based Delay Tolerant Network Formation in Heterogeneous 5G Using Inter-UA Collaboration. *PLoS ONE* **2016**, *11*, 1–37.
34. Neves, P.; Calé, R.; Costa, M.R.; Parada, C.; Parreira, B.; Alcaraz-Calero, J.; Wang, Q.; Nightingale, J.; Chirivella-Perez, E.; Jiang, W.; et al. The SELFNET Approach for Autonomic Management in an NFV/SDN Networking Paradigm. *Int. J. Distrib. Sens. Net.* **2016**, *2016*, 1–17.
35. 5G-NORMA Project. 5G NOvel Radio Multiservice Adaptive Network Architecture. Project Reference: 671584. Funded under: H2020-ICT-2014-2. Available online: https://5gnorma.5g-ppp.eu/ (accessed on 9 January 2017).
36. CHARISMA Project. Converged Heterogeneous Advanced 5G Cloud-RAN Architecture for Intelligent and Secure Media Access. Project Reference: 671704. Funded under: H2020-ICT-2014-2. Available online: http://www.charisma5g.eu// (accessed on 9 January 2017).
37. Xu, L.; Assem, H.; Yahia, I.G.B.; Buda, T.S. CogNet: A Network Management Architecture Featuring Cognitive Capabilities. In Proceedings of the 2016 European Conference on Networks and Communications (EuCNC), Athens, Greece, 27–30 June 2016.
38. Endsley, N.R. Design and Evaluation for Situation Awareness Enhancement. In Proceedings of the 32nd Annual Meeting on Human Factors and Ergonomics Society, Anaheim, CA, USA, 24–28 October 1988; Volume 32, pp. 97–101.
39. Sivarajah, U.; Kamal, M.M.; Irani, Z.; Weerakkody, V. Critical analysis of Big Data Challenges and Analytical Methods. *J. Bus. Res.* **2017**, *70*, 263–286.
40. Heijungs, R.; Henriksson, P.J.; Guinée, J.B. Measures of Difference and Significance in the Era of Computer Simulations, Meta-Analysis, and Big Data. *Entropy* **2016**, *18*, 361.
41. Holte, R.C. Very Simple Classification Rules Perform well on most commonly used Datasets. *Mach. Learn.* **1993**, *11*, 63–90.
42. Ditzler, G.; Roveri, M.; Alippi, C.; Polikar, R. Learning in Nonstationary Environments: A Survey. *IEEE Comput. Intell. Mag.* **2015**, *10*, 12–25.

43. Bouveyron, C.; Brunet-Saumard, C. Model-based Clustering of high-dimensional Data: A Review. *Comput. Stat. Data Anal.* **2014**, *71*, 52–78.
44. De Sanctis, M.; Bisio, I.; Araniti, G. Data Mining Algorithms for Communication Networks Control: Concepts, Survey and Guidelines. *IEEE Netw.* **2016**, *30*, 24–29.
45. Meng, W.; Li, W.; Kwok, L.F. EFM: Enhancing the Performance of signature-based Network Intrusion Detection Systems using enhanced Filter Mechanism. *Comput. Secur.* **2014**, *43*, 189–204.
46. Aggarwal, C.C. Outlier Analysis. Available online: http://www.charuaggarwal.net/outlierbook.pdf (accessed on 15 January 2017).
47. Katris, C.; Daskalaki, S. Comparing Forecasting Approaches for Internet Traffic. Expert Systems with Applications. *Expert Syst. Appl.* **2015**, *42*, 8172–8183.
48. Gardner, E.S.; Dannenbring, D.G. Forecasting with Exponential Smoothing: Some Guidelines for Model Selection. *Decis. Sci.* **1980**, *11*, 370–383.
49. Kadri, F.; Harrou, F.; Chaabane, S.; Sun, Y.; Tahon, C. Seasonal ARMA-based SPC Charts for Anomaly Detection: Application to Emergency Department Systems. *Neurocomputing* **2016**, *173*, 2102–2114.
50. Berlingerio, M.; Pinelli, F.; Calabrese, F. Abacus: Apriori-based Community Discovery in Multidimensional Networks. *Data Min. Knowl. Discov.* **2013**, *27*, 294–320.
51. Radenkovic, M.; Grundy, A. Efficient and Adaptive Congestion Control for Heterogeneous delay-Tolerant Networks. *Ad Hoc Netw.* **2012**, *10*, 1322–1345.
52. Zhang, T.; Wang, J.; Huang, J.; Huang, Y. Adaptive Marking Threshold Method for delay-sensitive TCP in Data Center Network. *J. Netw. Comput. Appl.* **2016**, *61*, 222–234.
53. Boem, F.; Ferrari, R.M.; Keliris, C.; Parisini, T.; Polycarpou, M.M. A Distributed Networked Approach for Fault Detection of Large-Scale Systems. *IEEE Trans. Autom. Control* **2017**, *62*, 18–33.
54. Venkatesan, R.; Er, M.J. A Novel Progressive Learning Technique for Multi-class Classification. *Neurocomputing* **2016**, *207*, 310–321.
55. Hayes-Roth, F.; Waterman, D.A.; Lenat, D.B. *Building Expert Systems*; Addison-Wesley: Boston, MA, USA, 1983.
56. Mas, M.; Monserrat, M.; Ruiz-Aguilera, D.; Torrens, J. RU and (U,N)-implications Satisfying Modus Ponens. *Int. J. Approx. Reason.* **2016**, *73*, 123–137.
57. Morsi, N.N.; Fahmy, A.A. On Generalized Modus Ponens with Multiple Rules and a Residuated Implication. *Fuzzy Sets Syst.* **2002**, *129*, 267–274.
58. Chen, H.; Li, T.; Luo, C.; Horng, S.J.; Wang, G. A Decision-Theoretic Rough Set Approach for Dynamic Data Mining. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 1958–1970.
59. Gilio, A. Generalizing Inference Rules in a Coherence-based Probabilistic default Reasoning. *Int. J. Approx. Reason.* **2012**, *53*, 413–434.
60. SELFNET Consortium. Deliverable 5.3: Report and Prototypical Implementation of the Integration of the Algorithms and Techniques Used to Provide Intelligence to the Decision-Making Framework. Available online: https://selfnet-5g.eu/2016/12/15/deliverables-online/ (accessed on 9 January 2017).
61. Talon, A.; Curt, C. Selection of Appropriate Defuzzification Methods: Application to the Assessment of Dam Performance. *Expert Syst. Appl.* **2017**, *70*, 160–174.

Orchestration of Use-Case Driven Analytics in 5G scenarios

Lorena Isabel Barona López, Jorge Maestre Vidal, Luis Javier García Villalba

Group of Analysis, Security and Systems (GASS)
Department of Software Engineering and Artificial Intelligence (DISIA),
Faculty of Computer Science and Engineering, Office 431
Universidad Complutense de Madrid (UCM),
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, Spain
E-Mail: lorebaro@ucm.es, jmaestre@ucm.es, javiergv@fdi.ucm.es

***Abstract*—** The SELFNET project provides an autonomic network management framework for 5G networks with a high degree of automation, self-healing and self-optimization. These capabilities are achieved through a layered architecture and a use-case driven approach. A differentiating feature on SELFNET is its competence when creating and customizing new use cases and their related virtual functions. In this way, the use case operators are able to introduce new rules and parameters that will be taken into account in the analysis and decision-making tasks. Due these characteristics, the orchestration of its analytical functions poses an important challenge in terms of configurability, synchronization and management of resources. In order to contribute to their resolution, this paper aims to lay the groundwork for implement the design and specification of the SELFNET Analyzer orchestration. To this end, several key issues related with the internal coordination of the analytics are introduced, among them initial assumptions, design principles, limitations, partitioning of the analysis process, data persistency and optimization. The proposed orchestration strategy has been implemented with different uses cases within the SELFNET Project.

Keywords – 5G, Situational Awareness; SDN/NFV; Data Analysis; Orchestration

1 Introduction

The amount and complexity of cyber threats have risen alarmingly in recent years [1]. Because of this, the information security management plays a very important role in the strategies of large organizations. Several guidelines and platforms for its implementation have been published (ISO/IEC 27000 [1], NIST-SP 800[3], CVSS-SIG-First [4], etc.), but despite its effectiveness in conventional scenarios, it has been shown that they do not adequately operate in dynamic monitoring environments [5]. This is the case of complex use cases, where the circumstances in which observations are made directly affect the ability of decision-making. In this context, examples of common issues when identifying the best mitigation/optimization actions are: inadequate asset assessment, fluctuations at data sources, difficulties when configuring new uses cases, and lack of scalability or interoperability.

In order to tackle these problems, there is a tendency to assume more cognitive methodologies, thereby facilitating understanding the environment through contextual analysis. High among those is the development of the Situational Awareness (SA) of the protected environment by applying the Endsley's Model [6]. In accordance with this method, the perception, comprehension and projection of the system status must be taken into account. As defined by M.R. Endsley, the term situational awareness refers to "the perception of the elements in the environment within a volume of time and space, comprehension of their meaning and the projection of their status in the near future", implicitly stressing how important the context is. As a result of the enormous complexity that entails managing the security of current networks, the Endsley's model has been specifically adapted to these scenarios, which has led to coining the term Network Security Situational Awareness (NSSA) [7].

Bearing this in mind, 5G networks, as clear examples of complex and dynamic monitoring environments are the focus of the research proposed in this paper. These technologies try to meet the requirements that are expected to be demanded by the current communication schemes in the short and long terms. As stated by A. Osseiran [8], they may be summarized in three great challenges: 1) enhancement of latency and reliability by supporting use-case dependent capabilities, such as the deployment of specific purpose applications, among them health-care, logistics, security or incidence response tools; 2) 5G must support a wide range of data rates with very high availability and reliability; 3) finally, in order to facilitate the inclusion of a large number of devices, networks must be scalable and flexible. Note that these endpoints must be simple enough to do not pose high battery consumption. In general terms, advances towards 5G technologies are based on combining and integrating a large number of emerging technologies, such as Network Function Virtualization (NFV) [9], Software Defined Networking (SDN) [10], Device to Device Communications (D2D) [11]; and analytic tools for network awareness, among them Artificial Intelligence (AI), Big Data or Self-Organized Networks (SON) [12].

At present, there are different projects aimed at facilitating the integration of these technologies into 5G scenarios. Significant efforts have been done by the European Commission under 5G-PPP and Horizon H2020 programs in order to support the new generation of mobile networks. It has led to the foundation of the 5G-PPP partnerships, which is committed to foster 5G advances in different strands such as cognitive network management or 5G Network Security [13]. Table 1 summarizes some of the projects involved in this association. Their differences and similarities are discussed in depth in [14]. Notable among them is the SELFNET approach [21], where an autonomic management framework to provide network intelligence and self-organizing capability for 5G mobile network infrastructures is provided.

Table 1: Research Projects on Mobile Networks

Project	Related Technologies	Use Cases
MCN [15]	SDN, Cloud Computing	1) Cloud Computing for mobile network operations, 2) end-to-end mobile Cloud
T-NOVA [16]	SDN, NFV	1)High-level scenario, 2) VNFs, 3) service chaining
UNIFY [17]	SDN, NFV	1) Infrastructure virtualization, 2) flexible service chaining, 3) network service chain invocation for providers
CROWD [18]	SDN, SON	General purpose
5G-NORMA [20]	SDN, NFV	1) multi-service, 2) multi-tenancy
CHARISMA [20]	SDN, NFV	General purpose
SELFNET [21]	SDN, NFV, SON, Cloud Computing	1) Self-healing, 2) Self-optimization, 3) Self-protection
COGNET [22]	SDN, NFV, Machine Learning	1) Situational context, 2) just-in-time services, 3) user-centric services, 4) optimized services, 5) SLA enforcement, 6) collaborative resource management
5G-ENSURE [34]	SDN, NFV, Security Models	11 Use Case clusters: 1-4) Identities, Authentication, Authorization and Privacy, 5) Software-Defined Networks, Virtualization and Monitoring, 6-10) Availability, Reliability and Integrity and 11) Lawful Interception.
SONATA [35]	SDN, NFV, cloud	1) Internet of Things, 2) Virtual CDN, 3) Guaranteed, resilient and secure service delivery in Industrial Networks, 4) vEPC, 5) Personal Security Applications, 6) Client and Hosting Service Providers,
5GNOW [36]	MTC, CoMP, M2M	1) PRACH scenario, 2) GFDM, 3) Uplink CoMP with joint reception, 4) Multiuser uplink on fragmented spectrum with FBMC, 5) Downlink CoMP with FBMC
METIS [37]	SDN, Multi-RAT, D2D, M2M	5 scenarios: 1) Amazingly fast, 2) great service in a crowd, 3) ubiquitous things communicating, 4) best experience follows you, 5) super real-time and reliable connections

SELFNET includes the widest variety of cutting-edge technologies and adapts the Endsley's Model [6], as well as the NSSA paradigm, to the 5G scene, as it is described in [23]. The latest effort toward providing SELFNET of an analytical component capable of meeting the 5G requirements in a use-case driven approach is summarized in [24], where the design principles, architecture and the formalization of how new use cases must be onboarded are detailed. But it does not explicitly indicate how all this information is organized, as well as how the analytical process is performed. The sophistication of these tasks results on the need of develop a novel orchestration of analytics (SELFNET Analyzer Orchestrator) , adapted to the 5G monitoring environment and the use-case driven politics derived from the SELFNET project, which is the main contribution of this paper. Other contributions are the specification and implementation of the dataflows in the

SELFNET Analyzer Framework, the proposal of strategies for their execution and optimization, and a battery of comprehensive examples which facilitates understanding the approach, and serves as a guide for the design and deployment of similar components at future projects. This paper is organized into eight sections, being the first of them the present introduction; Section II describes the essential elements of the SELFNET project, where the Analyzer and the specification of the onboarded data is emphasized; Section III introduces the initial assumption of the SELFNET orchestrator; Section IV explains its design principles; Section V details its workflow; Section VI discusses the execution and optimization strategies; Section VII illustrates a battery of practical examples; Finally, Section VIII concludes this work.

2 Background

This section describes in detail the key points of SELFNET necessary for understanding the Analyzer and its orchestration. In particular, the SELFNET architecture and its adaptation to the NSSA, the design of its Analyzer Module and the descriptors of the use cases are reviewed.

2.1 SELFNET and the Situational Awareness on 5G Scenarios

SELFNET H2020 Project provides a smart autonomic network management framework for 5G mobile networks based on the combination of 5G key-enabled technologies: SDN, SON, NFV, Artificial Intelligence and cloud computing. SELFNET enables the autonomic deployment of virtual network functions and the reconfiguration of network parameters in order to mitigate existing or potential problems, while maintaining the Quality of Experience (QoE) of end users [21]. These capabilities are provided by means a layered architecture and a use-case driven approach. On the one hand, three use cases were defined: i) self-protection capabilities to mitigate or prevent security problems such as a cyber-attack, i) self-healing capabilities to prevent or correct network failures and iii) self-optimization to dynamically improve the service and network performance. For this purpose, SELFNET proposes two kind of advanced network functions: i) sensors to monitor specific network information and ii) actuators to perform countermeasures to fix or mitigate possible problems. On the other hand, SELFNET architecture is based on six layers (Fig. 1): Infrastructure Layer, Data Network Layer, SON Control Layer, SON Autonomic Layer, NFV Orchestration & Management Layer and SON Access Layer, as is described in [38].

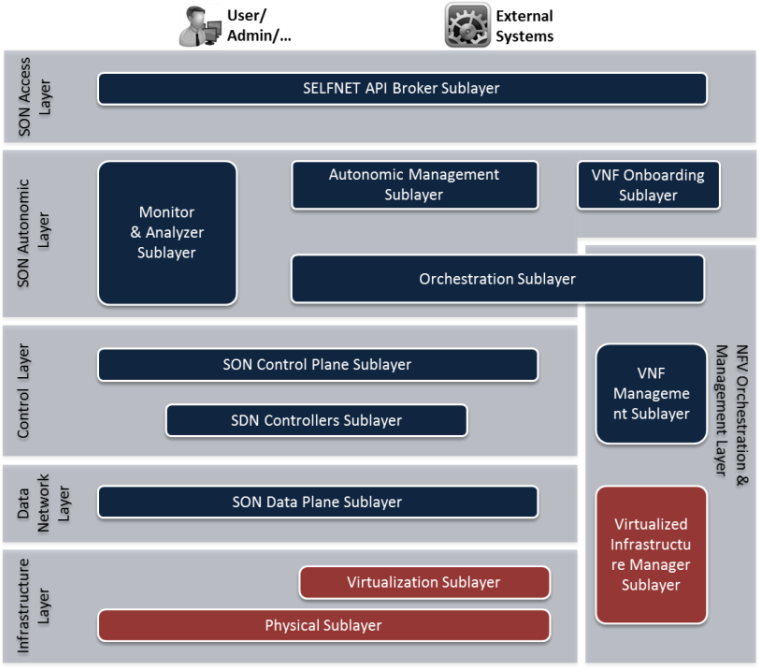

Figure 1. SELFNET Architecture

- Infrastructure Layer. It provides the physical resources required for the instantiation of virtual functions. The Physical Sublayer, Virtualization Sublayer and Cloud Computing Sublayer enable the virtualization of compute, network and storage resources.
- Data Network Layer. The network functions (NFs) are instantiated and interconnected in a designed topology. It includes the NF required for normal operation and SON functionalities.
- Control Layer. It includes the SON sensors and actuators. The SON sensors collect data from different sources and the SON actuators execute response actions into the network. These elements are controlled by the SON Autonomic Layer (intelligence).
- SON Autonomic Layer: This layer is responsible for providing the network intelligence. For this purpose, the system monitors and analyse the incoming information in order to diagnosis network problems. Then, it uses the available network functions to decide the best reaction strategy. Taken decisions are sent to NFV orchestration and Management Layer.
- NFV Orchestration & Management Layer. It controls the deployment and instantiation of the different NFs in the infrastructure. This layer follows the ETSI MANO recommendations.
- SON Access Layer. It provides the interface used by external actors like Business Support Systems (BSS) or Operational Support Systems (OSS). Similarly, the network administrator also can stop, verify and enforce actions on SELFNET.

In turn, SON Autonomic Layer is responsible to provide the network intelligence by means Monitor and Analyzer sublayer and Autonomic Management sublayer. In particular, the Situational Awareness of SELFNET is achieved through the application of Endsley Model [6], which define three main phases: Perception (Monitor), Comprehension (Aggregation) and Projection (Analysis and Diagnosis) as is shown in Fig 2.

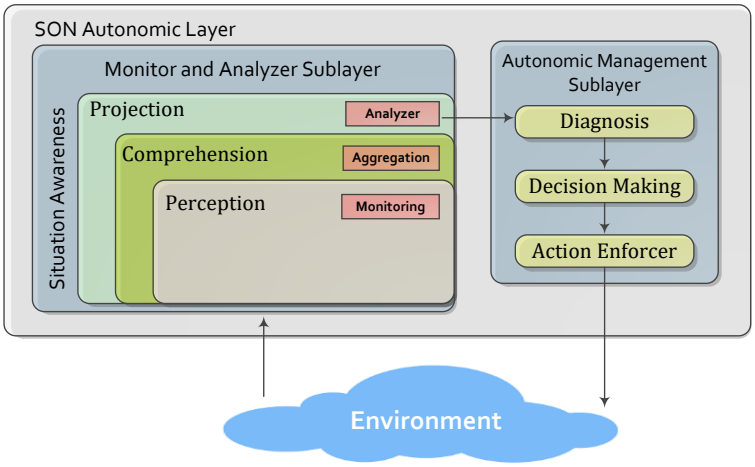

Figure 2: Situational Awareness in SELFNET Project

This approach supposes a high challenge because the information is gathered from different sources (Monitoring task) and then the raw data is aggregated and correlated in order to provide high level metrics (aggregation and correlation task). In the next step, suspicious conditions are inferred or detected (Analyzer task) and then they are sent to Diagnosis sublayer. Finally, this sublayer applies advanced intelligent techniques to perform proactive and reactive actions.

2.2 SELFNET Analyzer: Design Principles and Architecture

The general assumptions, requirements and the first items to consider related with the design principles of the Analyzer were previously introduced in [24]. In accordance with this publication, it must be 1) scalable, extensible and multi-level by design; 2) use-case driven, where the use-case operators are able to specify the inclusion/modification of its functionality; 3) the use case knowledge-bases required for the analytics are provided by skilled operators or by accurate machine learning algorithms; 4) user-friendly in terms of use case declaration and composition of knowledge inference rules; 5) the management of knowledge considers uncertainty and stochastic events; 6) the data sources do the filtering of the input data, hence removing inconsistencies, ambiguity and repetition on the crisp data (i.e. the SELFNET Analyzer does not perform filtering actions). All these assumptions and limitations are inherited by the orchestrator, and therefore they are considered in this proposal.

The SELFNET Analyzer relationship with the rest of the project components is summarized in Fig. 3, where a view of their main data sources is illustrated as a black box model. There two main information sources as facts Fa, were identified: Aggregation (Events $Fa(Ev)$, Thresholds $Fa(T_H)$ and Key Performance Indicators $Fa(KPI)$) and internal analytic elements (pattern recognition $Fa(PR)$, forecasts $Fa(Ft)$ and adaptive thresholds $Fa(AT_h)$). The final conclusions that compose the SELFNET Situational Awareness are sent to Diagnosis module labeled as symptoms, via reports.

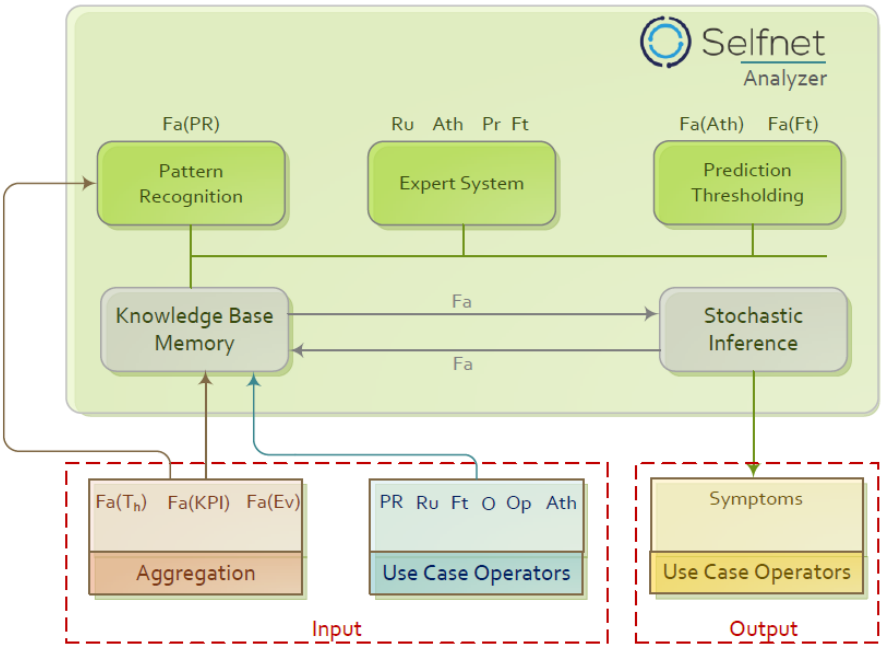

Figure 3: Inputs and Outputs on SELFNET Analyzer

The SELFNET Analyzer architecture is shown in Fig. 4. It is centralized and their components are divided into eight main elements: Pattern Recognition (no. 1), Prediction (no. 2), Adaptive Thresholding (no. 3), Knowledge-base (no. 4), Inference Engine (no. 5), Memory (no. 6), User Interface (no. 7) and Uncertainty Estimation (no. 8). Where Pattern Recognition infers new facts related with patterns and regularities found in the aggregated data, Prediction discovers facts related with forecasting aggregated data or previously known facts, and Adaptive Thresholding establishes the limitations to be taken into account when inferring new knowledge. The core of the SELFNET Analyzer is a rule base engine composed by the Knowledge-base, Inference Engine and Memory. It applies use-case driven rules for deducting conclusions from the previously identified facts. If some of them match with situations of interest for the Diagnosis module, they are adapted by the Uncertainty Estimation component, which allows them to be interpreted as symptoms by the SELFNET upper layers. Note that the configuration of the use cases is performed at the User Interface.

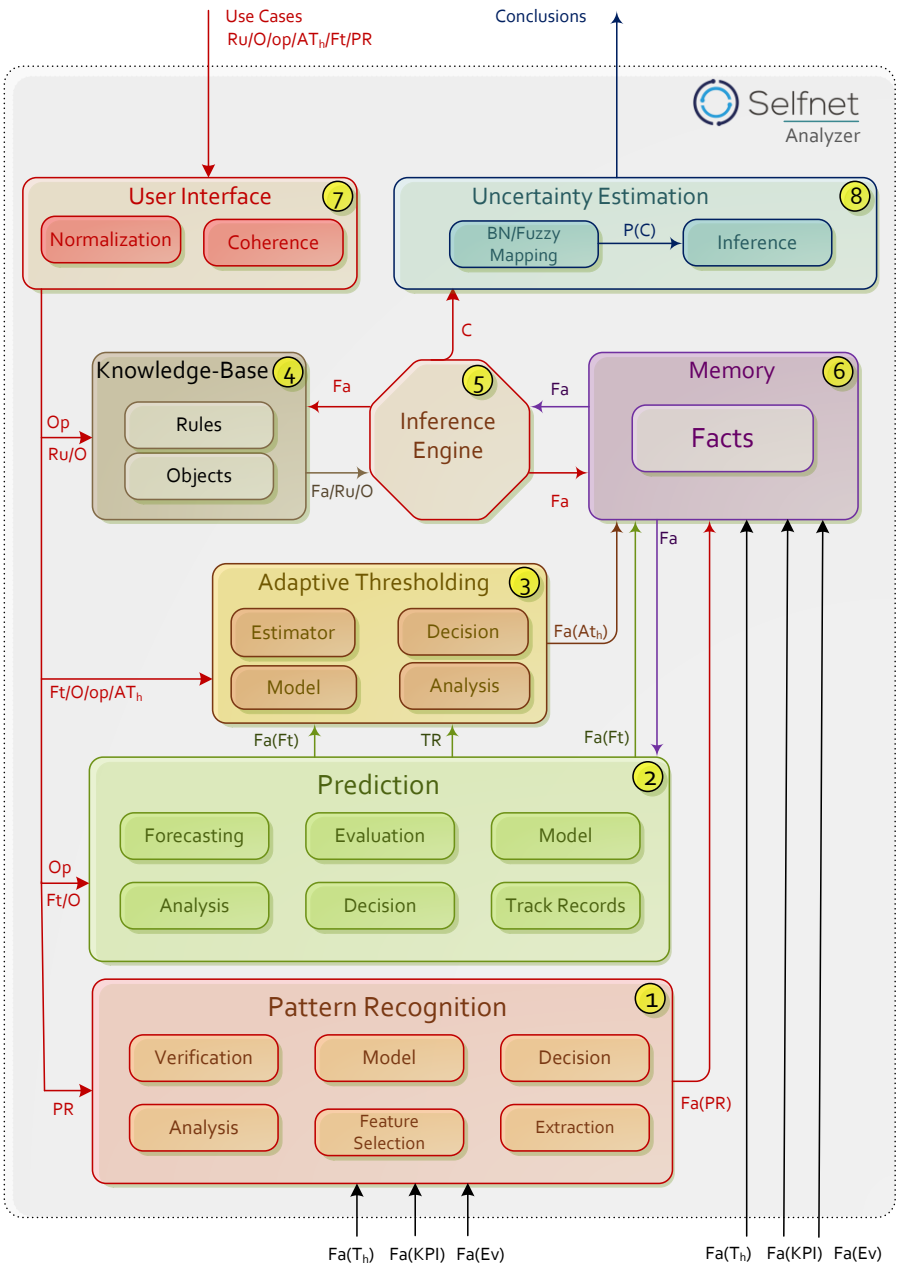

Figure 4: SELFNET Analyzer Architecture [24].

2.3 Specification of the Use Cases

When initiated, the SELFNET Analyzer is a *tabula rasa* without actions nor reasoning to be orchestrated. It requires the onboard of use cases, which provides the script with the activities that may be performed. If a new use case is onboarded, the information that it is able to manage, as well as the analytic actions which might be executed, are specified according to the descriptors summarized in Table 2.

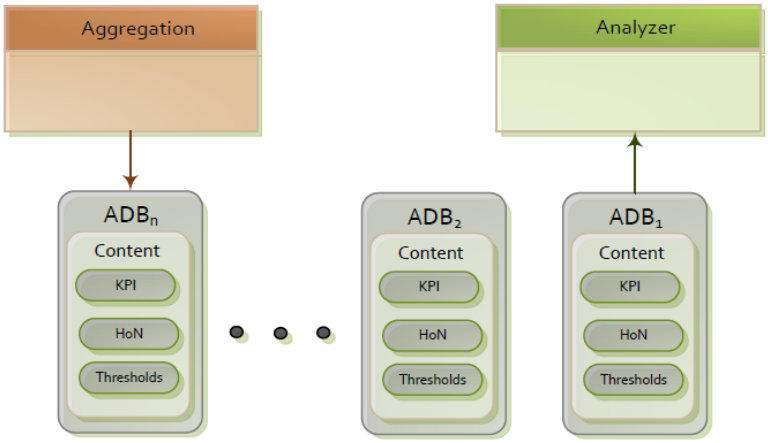

Figure 5: Communication by ADBs

Table 2: Summary of Use Case Data Specification

Data	Category	Provider	Destination	Format
Object (simple) O	Specification	Use Case	Analyzer	O_i:{*object name*|*weight*|*noValues*|*range of values Va*}
Object (mult) O	Specification	Use Case	Analyzer	O_i:{*Object name*|*weight*|*noValues*|$[Va_1][Va_2]$ … $[Va_K]$}
Operation Op	Specification	Use Case	Analyzer	Op_i:{*name*|*symbol*|*priority*|*operands*|*description*}
Facts Fa	Assessment	Aggregation Analyzer	Analyzer	Fa_i:{*expression*|*eight*|*uncertainty*|*timestamp*|*location*}
Rule Ru	Specification	Use Case	Analyzer	Ru_i:{*rule*|*priority*|*use case*}
Forecast (ts) Ft	Specification	Use Case	Analyzer	Ft_i:{*timeSeries*|*object*|*domain*|*lenght*}
Forecast (G) Ft	Specification	Use Case	Analyzer	Ft_i:{*graph*|*object*|*noVertex*|*domain*|*lenght*}
Threshold T_h	Specification	Use Case	Analyzer	T_{hi}:{*Th name*|*object*}
A. Threshold AT_h	Specification	Use Case	Analyzer	Ft_i:{*ATh name*| *data structure*|*CI*|*forecast*}
Datasets D	Specification	Use Case	Analyzer	D_i:{*D name*|*object*|*type*|*source*}
Pattern Recognition	Specification	Use Case	Analyzer	PR_i:{*PR name*|*objectIn*|*objectOut*|*action*|*reference data*
Conclusion C	Specification	Use Case	Analyzer	St_i:{*C name*|*use case*|*fact*}
Report Re	Report	Analyzer	Diagnosis	Re_i:{*C name*|*use case*|*fact*|*uncertainty*|*trigger*}

The objects O describe the nature of the data to be analyzed and the elements from which the rule-based expert systems infers knowledge. Operations Op establish binary relationships between facts. Thresholds T_h are delimitations calculated at the Aggregation layer. Facts Fa are basic elements of the SELFNET reasoning which describe how the Analyzer Module acquires new knowledge via its rule-based expert system. Rules Ru indicate how the SELFNET Analyzer infers new facts at the rule-based expert systems. Note that they are propositional logic expressions in *modus ponens* where the implications deduce the new knowledge. Forecast Ft, pattern recognition PR and adaptive thresholds AT_h specify the basic analytical operations, for which the datasets D provide additional collections of reference samples. Finally, conclusions C state facts related with symptoms.

3 Assumptions

The orchestration of the Analyzer accepts the assumptions and limitations established in [24], which were described in the previous section. In order to satisfy the needs of the previously agreed design, as well as to be able to provide the functionalities expected by the rest of the SELFNET tasks, it additionally identifies the following new specific constraints to be considered.

3.1 Symptoms and Events

The Diagnosis layer of SELFNET [38] distinguishes two groups of reports: symptoms and events [23]. The first one contains conclusions generated through analytics. On the other hand, events are signals on which it is not necessary to carry out actions related to Artificial Intelligence, such as pattern recognition, prediction or logical inference. Note that in [24], events were managed as facts (in particular $Fa(Ev)$). In the same way as the rest of the metrics extracted from the aggregated information (see Fig. 4), events were included in the working memory, and hence they could be considered for acquiring knowledge via rule-based expert system, forecasted or studied by pattern recognition techniques. Obviously this was a potential contradiction that must be clarified. In the remainder of this paper, it is assumed that the expression $Fa(Ev)$ strictly refers to aggregated metrics extracted from the monitored events, instead of the event themselves. For example, alerts issued by the IDS involved in the use case Self-Protection are, by definition, events. Given their relevance, they must be directly addressed to the Diagnosis layer, so it is not possible to assume the cost in time that involves the execution of complex analytical calculations on them. However, it is possible to generate metrics that facilitate the making of future decisions or even foresee the issuance of new alerts. For example, the Aggregation layer may provide information about the number of alerts per observation, mean, variance, emission intervals, and its distribution, among others. From which stronger conclusions could be inferred. Unlike when dealing with events, these metrics are not processed with enough efficiency to deliver real-time results.

3.2 Rule based Inference

Given the SELFNET framework and the nature of the monitored data, the decision to implement a rule-based inference engine as a symptom discovery tool brings many benefits, among them: 1) rule engines allow to use case administrators decide "What to do", not "How to do it". Because of this, it makes it easy to express solutions to difficult problems and specify the onboard of future use cases. 2) It brings logic and data separation, where data is in the domain of objects, and the logic is in the rules Ru. 3) It provides centralization of the knowledge required for infer symptoms. 4) Rule-based systems are fast and scalable: some algorithms (ex. RETE, Leaps, Treat, etc.) [25] and their optimizations [26] provide very efficient ways of matching rule patterns to the use cases domain object data. These are especially efficient when facts change in small portions as the rule engine can remember past matches. For example, this happens with the information periodically provided by a particular SELFNET sensor. But rule-based systems also pose drawbacks: the first of them is high dependency of the rule set. If the rules are not consistent, coherent or reasonably specific, the results obtained will be probably not as expected [27]. On the other hand, they are susceptible to bad practices. For example, rule-based systems allow storing, managing and updating rules as data. It is common that they are mistakenly used to generate new rules or even update them at runtime, which is out of the scope of these technologies. Finally, it is important to bear in mind that the scalability of rule-based systems has a negative impact in terms of resource consumption. In this regard, it is worth mentioning the consequences of their two most frequent ways to scale [28]: firstly, if the number of facts is acceptable, but the number of rules is very high, there will be an important increase in the computation time of their processing. On the opposite, if the number of facts is very high, but the number of rules is acceptable, a larger amount of memory is required for storage. Note that if the number of inputs and rules are large, then both, memory and efficiency are penalized. In the context of SELFNET it is expected to receive a large number of facts, but operate on small rule sets. Consequently, it is expected that the scalability of the expert rule-based system will lead to the use of a greater amount of storage space.

3.3 Data Granularity

SELFNET is a complex monitoring scenario where a large amount of sensors collect information about the state of the network in real time. This information is processed in the aggregation layer, which provides the necessary metrics to acquire knowledge. For this purpose, the Analyzer must perform complex calculations. As will be described in the later sections, aggregated data will not be raw processed. Instead, it will be packed as Aggregated Data Bundles (ADB) which will periodically be loaded by the Analyzer and converted into facts. Each ADB is the summary of all the system information observed over a time period T. It can therefore be stated that ADB may be abstracted as an observation on a time series of records that facilitate the network

awareness. It is assumed that the effectiveness and performance of the analytics depends on the T, and how representative is the information on the ADB.

4 Design Principles

The following design principles and limitations lay the foundation of the Analyzer orchestrator, as well as the implementation of its internal components, data flows and synchronization.

4.1 Aggregated Data Bundles

The information required for the analytics is obtained from the Aggregation layer packaged as Aggregated Data Bundles (ADB). An ADB is the summary of the aggregated metrics calculated in a time interval P translated into facts Fa. Note that a priori, the data within an ADB does not overlap the metrics on other ADBs (this aspect could be revised later for future optimizations). For example, let the time series $Y = \{Y_t : t \in T\}$ where $Y_1, Y_2, \dots, Y_k$, $k = 7$, assuming the construction of ADBs on $P = 1$, the SELFNET Analyzer will sequentially deal with 7 ADBs, i.e. $ADB_1, ADB_2, \dots, ADB_k$ (see Fig. 5). Through the use of this strategy a massive and continuous input of information is avoided, which facilitate the initialization of the implemented data mining algorithms. Likewise, the information is managed and processed in an orderly manner, which also reduces the number of inconsistencies between the new facts and the data stored in the working memory. Finally, as is illustrated at the next section, the deployment of optimization method based on the exploitation of concurrence is facilitated.

4.2 Persistence

The SELFNET Analyzer does not provide persistence of the data loaded as ADBs. The monitored raw data and aggregated metrics are conveniently stored in the Big Data platform located at the Aggregation layer. Facts Fa not implicated in prediction/pattern recognition are discarded once their ADB is completely processed and the conclusions are inferred. This means that, in this case, facts Fa are temporally stored in a local short-term memory only for the duration of their analysis. On the other hand, facts Fa required for prediction/pattern recognition may temporally persist throughout the analysis of various ADBs. This is because they compose the time series and graphs needed to build models/regressions. Note that these data structures have limited size, which once reached involves eliminating the more obsolete observations via First In First Out (FIFO) policies [29]. Once an ADB is completely analyzed and the conclusions are reported to the Diagnosis layer as symptoms, the working memory of the rule-based expert systems is restarted. Only the necessary facts for the construction of the time series and graphs are temporarily conserved, but this is outside the working memory. When loading a new ADB, facts on time series and graphs are again, added to the working memory as $Fa\ (Ft)$, $Fa\ (Ath)$ and $Fa(PR)$.

4.3 Analytic Pipelining

Analytics are executed as a linear pipeline of sets of data processing elements connected in series, where the output of an input is the input of the next one [30]. When an ADB reach the SELFNET Analyzer, a sequence of processing elements is executed, where intelligence actions (i.e. logic inference, pattern recognition, prediction) and preprocessing steps (load ADBs, data encapsulation, generation of reports) are chronologically separated, and their inputs/outputs are shared by buffer storage structures. So it is possible to state that this first approach considers a buffered-synchronous pipeline analytic architecture. Its main advantages are: great organization of information to process, mitigation of inconsistencies between the new facts and the data being analyzed, easy of design and modularity. The latter allows managing every set of actions independently, which facilitates debugging, troubleshooting tasks and provide a more accurate assessment of the performance of their analytic actions. But it is important to keep in mind that this scheme also poses several challenges, among them try to define sets of actions of similar complexity in order to enable optimization strategies based on parallelism, the fact that the delay in a task may slow down the execution of those that depend on it, and in the case of implement parallelism, the best suited politics of temporal memory sharing must be identified.

5 Workflow

The SELFNET Analyzer orchestration is separated into seven main steps: use case Onboarding [O], Discovery [DIS], Patter Recognition [PR], Prediction [FT], Adaptive Thresholding [ATH], Knowledge inference [KI] and Notification [N]. They are illustrated in Fig. 6 and described in detail bellow.

- *Onboarding [O]*. The onboarding step is executed only once per use case. It corresponds to the component User Interface in [24], and allows updating the knowledge-base by inserting, modifying or deleting data associated with every use case, such as objects O, rules Ru operations Op or prediction metrics Ft. When a new use case is onboarded, the input data is normalized, and in order to avoid runtime errors, the coherence of the new specification is validated. Then the Analyzer is prepared to accommodate the new operations, hence including the specified information on the existing data structures, memory allocation and synchronization of the onboarded actions with the previous loaded configurations.
- *Discovery [DIS]*. The discovery step is the link between the SELFNET Aggregation and Analyzer layers. These tasks periodically receive ADBs which summarize the SELFNET aggregated observations. From the loaded KPI, events and thresholds, the Analyzer build facts ($Fa(KPI)$, $Fa(Ev)$ and $Fa(Th)$). If they are required for prediction, patter recognition or adaptive thresholding, the Analyzer includes these observations in the temporally stored time series or graphs. Note that independent facts are removed at the end of the ADB processing, as well as the new knowledge acquired from them.

- *Pattern Recognition [PR]*. The set of actions related with pattern recognition implies the access to the datasets with models, sample collection or signatures, and the detection of matches or outliers. The acquired facts may be considered by prediction, pattern recognition or adaptive thresholding, as well as to infer knowledge on the rule-based expert system.
- *Prediction [FT]*. The set of actions related with prediction includes the construction of forecasting models/regression, the decision of the best suited algorithms by considering the nature of the input data, and the estimation of its evolution. As is the case on the pattern recognition activities, the generated facts may be considered to infer knowledge on the rule-based expert system, and also to identify adaptive thresholds.
- *Adaptive Thresholding [ATH]*. This set of operations establishes measures to approximate when the forecasting errors must be taken into account when identifying symptoms. In order to enhance the information reported to the Diagnosis layer, the new facts are provided to the rule-based expert system, hence contributing to the inference of new knowledge.

- *Knowledge inference [KI]*. This step executes the tasks related with the rule-based expert system. It considers the data provided by the sources of information mentioned above, among them facts directly built from aggregated data, pattern recognition, prediction and adaptive thresholding steps. The acquired knowledge is included in the SELFNET Analyzer working memory. Conclusions are transmitted to the notification capabilities as potential symptoms.
- *Notification [N]*. The set of actions on Notification corresponds to those on the component Uncertainty Estimation at the original SELFNET Analyzer architecture. They are the link between the SELFNET Diagnosis layer and the knowledge acquired by the Analyzer. This step performs two main groups of tasks: accommodation and formatting. The first one filter redundant and low representative information. Once the ADB is completely analyzed, these actions erase and restart the auxiliary functionalities on the analytics and the several data structures; only the information required for build time series and graphs from data included in future ADBs is temporally persistent. On the other hand, the group of actions related with formatting, translates internal information of the analyzer to crisp data required by Diagnosis. Then it is reported.

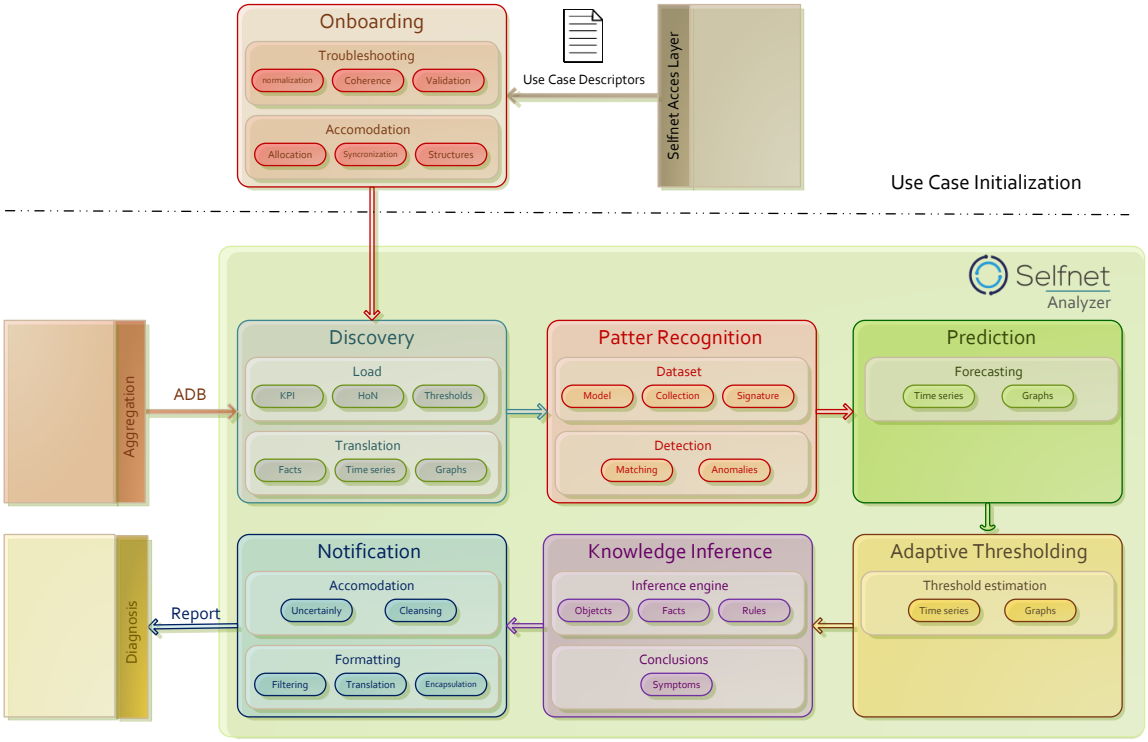

Figure 6: Sets of Actions on the Analyzer

6 Execution and optimization

When no optimization measures are implemented, the execution of the sets of actions determined in the previous section can be summarized in Fig. 7. There the onboard of a new use case and the completion of its different task are illustrated. Note that in accordance with this basic specification, the Analyzer only is able to load a new ADB if the previously loaded ADB is completely processed. Obviously this is not the most efficient way to carry out their study. Assuming separately the computational costs of every set of actions: $O(DIS)$, $O(PR)$, $O(FT)$, $O(ATH)$, $O(FT)$, $O(KI)$ and $O(N)$; and ignoring the penalty of onboarding use cases $O(Onboard)$, the average cost of analyze an ADB is:

$$O_{ADB} = O(DIS + PR + FT + ATH + KI + N)$$

Where given the complexity of the pattern recognition and prediction methods, $O(FT)$ and $O(ATH)$ will concentrate most of the resource penalty. This approach is cheap in terms of memory, because once a set of actions is completed, most of their auxiliary data structures can be released. In addition, not managing different ADBs in parallel prevents the replication of such containers. Because of its simplicity and easy debugging, this is the first version of the Analyzer orchestrator that was implemented.

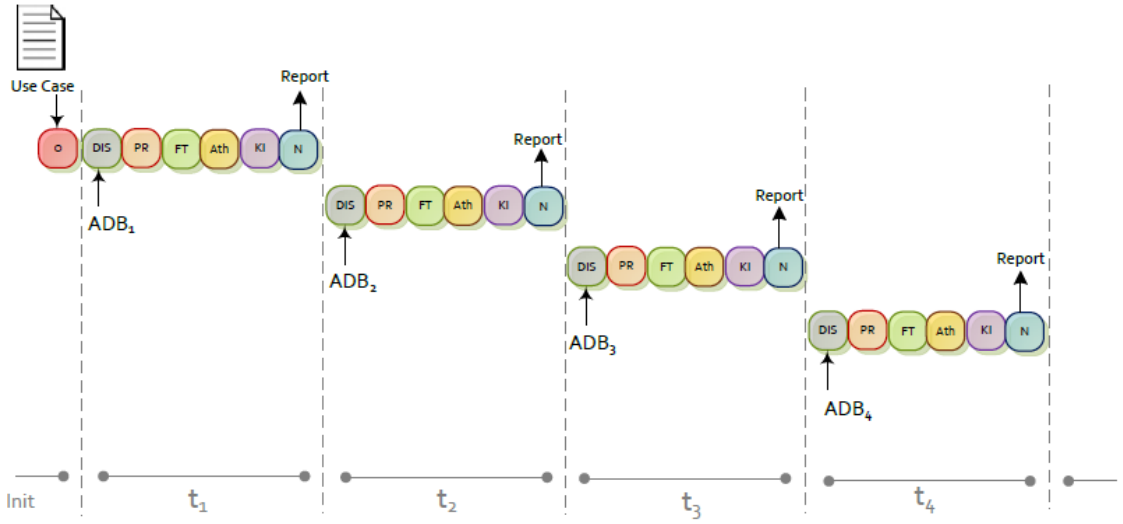

Figure 7: Basic Execution of the Analyzer sets of Actions

However, this scheme can be optimized easily by considering pipelining solutions. They allow overlapping execution of multiple actions with the same memory space by exploiting parallelism [31]. Fig. 8 illustrates an example of these kinds of methods, where 6 ADBs can be processed at the same time period.

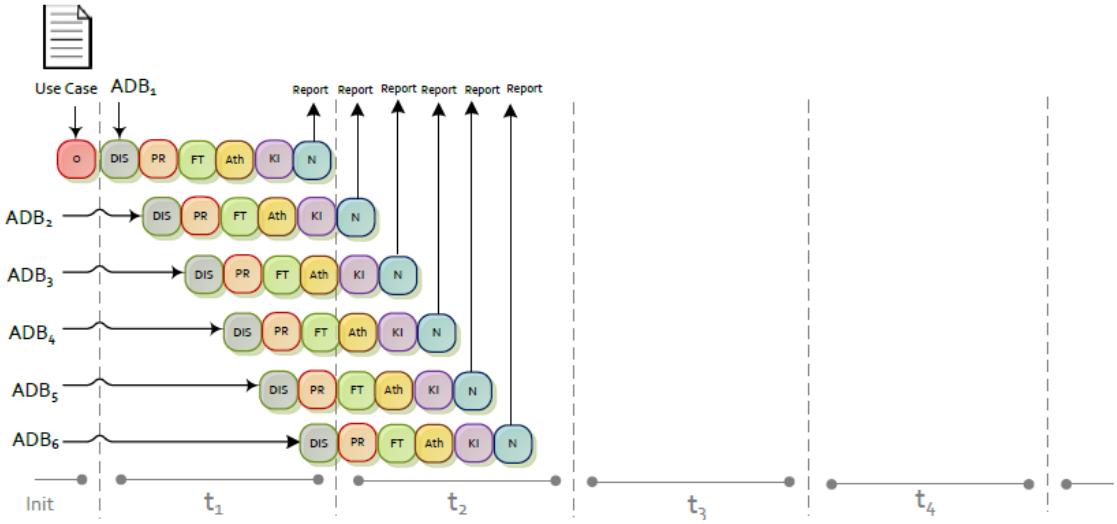

Figure 8: Example of Optimal Analysis of Multiple ADBs in Concurrency

Two sets of similar actions cannot be processed in concurrency, but it is possible with different sets. This means that, for example Patter Recognition on the analysis of ADB_1 cannot overlap with Pattern Recognition on the following ADB_2, since all the resources for this task are being used to analyze the first information package. But it could be executed in concurrency with the Discovery stage of ADB_2, where resources and memory are not shared. If the initialization cost related with processing the first ADB is ignored, it can be formalized as follows:

$$O_{init} = O(DIS + PR + FT + ATH + KI + N)$$

Then the cost of analyze ADB_s at $init + 5$ is summarized as

$$O_{ADB} = O(MAX\{DIS, PR, FT, ATH, KI, N\})$$

This implies an important improvement over the original proposal. But the implementation of this scheme leads to several restrictions. Firstly, it requires a greater amount of memory; the system must support up to six times more storage space to facilitate the analysis of six ADBs at a time. On the other hand, in order to allow the communication between sets of actions, the SELFNET Analyzer must provide temporal storage buffers and synchronization mechanism. This requires managing shared memory between tasks, and adds complexity to the execution thread. Furthermore, it has to be borne in mind that under optimal circumstances, all sets of actions must take the same time to complete. Obviously this does not happen in reality, since pattern recognition and prediction actions often imply a higher cost than those relate with the rule-based inference. Consequently, it is possible that certain sets of actions must remain on hold until others are finished, before giving way to new analysis processes. This problem is illustrated in Fig. 9, where the different sets of actions display unequal time consumption. If there are no waits, the different tasks will overlap leading to memory-sharing conflicts and inconsistencies between facts. For example, ADB_3 prediction actions require the pattern recognition facts of the same processing thread. But if such overlapping occurs, prediction on ADB_3 may also receive facts derived from pattern recognition at ADB_2, which would lead to inference erroneous knowledge. Note that it is also possible that none of the use cases require the execution of some sets of actions (in Fig. 9 this occurs with adaptive thresholding tasks). In both circumstances there will be moments of waiting.

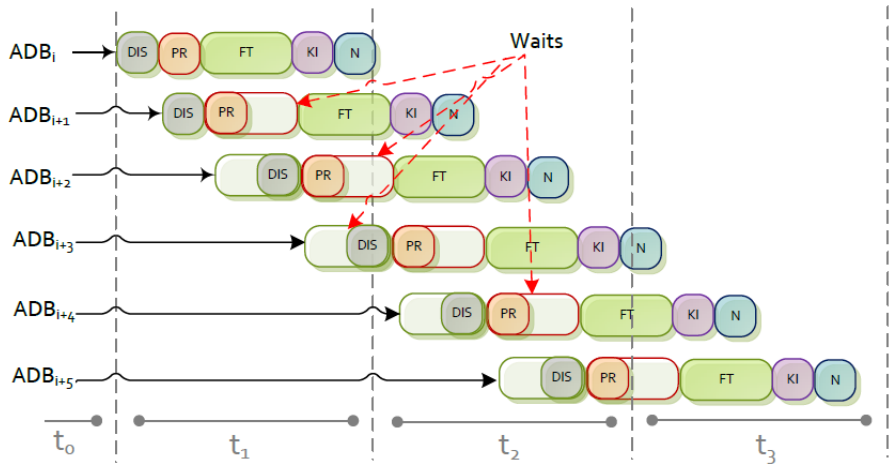

Figure 9: Example of Computational time Penalization because of Unequal Set of Actions

Another clear example of inequality between execution costs of sets of actions is shown in Fig. 10, when the same task, in this case prediction, becomes more and more expensive over time. This entails an accumulative delay in the previous actions (pattern recognition).

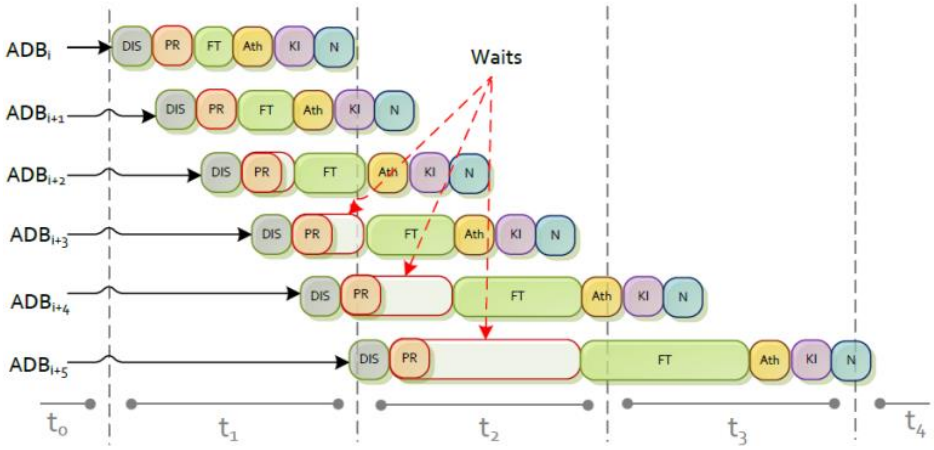

Figure 10: Example of computational time penalization because of incremental resource consumption

The Analyzer orchestrator deals with these problems by adjusting the granularity of the information provided by the ADBs, and by limiting the observation sliding windows and the amount of information considered for initializing the pattern recognition and prediction algorithms. According to these circumstances, the cost of executing an ADB_s at $init + 5$ once the sequencing is initialized is expressed as follows:

$$O_{ADB} = O(MAX\{DIS, PR, FT, ATH, KI, N\}) + delay_t$$

Where the cumulative penalization is decomposed as:

$$delay_t = O(Wait_{DIS} + Wait_{PR} + Wait_{FT} + Wait_{ATH} + Wait_{KI} + Wait_N)$$

Alternatively, each set of actions is also able to exploit concurrency at thread level in order to improve its performance. Due to the characteristics of the monitoring environment, it is possible to deduce that frequently, the same metric (ex. temperature, congestion, etc.) may be reported from different sources. The analysis of similar information, but provided from different data sources, is enhanced by CPU/GPU multithreading [32] as it is illustrated in Fig. 11, which improves their consumption of computational resources in terms of storage and efficiency.

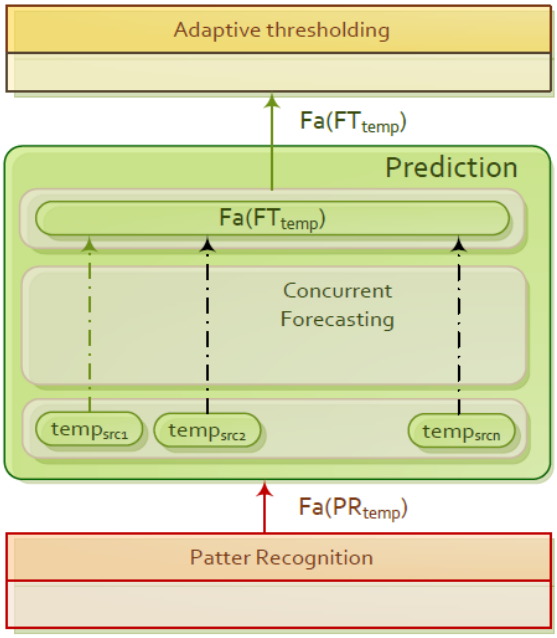

Figure 11: Example of Concurrency Exploitation

7 Illustrative Examples

This section describes several examples of the analytic process according with the aforementioned Analyzer Orchestration scheme.

7.1 UC 1: Device Packet Loss

7.1.1 Description

The illustrative use case called $Self - packet\ loss\ prevention$ (SLP) reports symptoms related with huge packet loss rates on SELFNET devices. Where if the packet loss rate of certain SELFNET device exceeds a specific threshold, a new fact that represents such situation is acquired. This is a very basic example where concurrency pipelining is not applied, and where prediction and adaptive thresholding are not considered.

Therefore the decision thresholds are static and were built at Aggregation. Table 3 shows its onboarding descriptors according with the specification summarized in Table 2.

Table 3: ***SLP*** Onboarding Specification

Item	Descriptor
Object	O_1:{$Packetloss$|1|1|$\mathbb{R}$}
Threshold	T_{h1}:{$maxPacketLoss$|O_1}
Operator	Op_1:{$Equal$| = | 1 |$(Fa, O, Va) = (Fa, O, Va)$|equal}
Operator	Op_2:{LGT| ≥| 1 |$(Fa, O, Va) \geq (Fa, O, Va)$|left is GE}
Conclusion	C_1:{$excessive\ packet\ loss$|SLP|$Fa(O_1) \geq Fa(T_{h1})$}
Rule	Ru_1:{$Fa(O_1) \geq Fa(T_{h1}) \rightarrow Fa(C_1)$|1|$SLP$}

7.1.2 Step-by-Step

The following illustrates and example of runtime in $Self - packet\ loss\ prevention$, where different ADBs are loaded and analyzed according to the aforementioned indications. Fig. 12 displays every step in a sequence diagram, which are described step-by-step below:

1. The SLP use case descriptors are loaded by the SELFNET Analyzer. Then, the memory for storing temporal containers of objects O_1, thresholds T_{h1} and facts Fa is allocated. Pattern recognition, prediction and adaptive thresholding are not required, so neither data structures to support time series nor graphs are considered.

2. The ADB_1 with aggregated instances of $O_1$and T_{h1} is requested to the Aggregation layer and then it is processed. The SELFNET Analyzer build the following facts from the information gathered by the network elements ($NodeA$, $NodeB$, $NodeC$, $NodeD$):

$$Fa(O)[idO1]{:}\{O_1 = 0.35|1|1|Today\ 12{:}22{:}15|NodeA\}$$
$$Fa(O)[idO2]{:}\{O_1 = 0.34|1|1|Today\ 12{:}22{:}15|NodeB\}$$
$$Fa(O)[idO3]{:}\{O_1 = 0.33|1|1|Today\ 12{:}22{:}15|NodeC\}$$
$$Fa(O)[idO4]{:}\{O_1 = 0.35|1|1|Today\ 12{:}22{:}15|NodeD\}$$

   And by the data Aggregation:

$$Fa(Th)[idTh1]{:}\{T_{h1} = 0.7|1|1|Today\ 12{:}22{:}15|All\}$$

3. Given that pattern recognition, prediction and adaptive thresholding are not required, the Analyzer bypasses those steps (i.e. new facts are not inferred by them)
4. The rule-based inference engine processes the discovered facts by applying the rule Ru_1. Given that the condition $Fa(O_1) \geq Fa(T_{h1})$ is not satisfied by any of the facts, conclusions related with SLP are not inferred.
5. All the temporal data related with objects O_1, thresholds T_{h1} and facts Fa is cleaned.

6. The ADB_2 with aggregated instances of $O_1$and T_{h1} is requested to the Aggregation layer and processed. The SELFNET Analyzer build the following facts from the information gathered by the network elements ($NodeA$, $NodeB$, $NodeC$, $NodeD$):

$$Fa(O)[idO5]{:}\{O_1 = 0.36|1|1|Today\ 12{:}23{:}15|NodeA\}$$
$$Fa(O)[idO6]{:}\{O_1 = 0.34|1|1|Today\ 12{:}23{:}15|NodeB\}$$
$$Fa(O)[idO7]{:}\{O_1 = 0.81|1|1|Today\ 12{:}23{:}15|NodeC\}$$
$$Fa(O)[idO8]{:}\{O_1 = 0.31|1|1|Today\ 12{:}23{:}15|NodeD\}$$

   And by the data Aggregation:

$$Fa(Th)[idTh2]{:}\{T_{h1} = 0.79|1|1|Today\ 12{:}23{:}15|All\}$$

7. The Analyzer bypasses pattern recognition, prediction and adaptive thresholding.
8. The rule-based inference engine processes the discovered facts by applying the rule Ru_1. Given that the condition $Fa(O_1) \geq Fa(T_{h1})$ is satisfied for the data gathered by $NodeC$, The following fact related with $Self - packet\ loss\ prevention$ is inferred.

$$Fa[\text{idF1}]:\{Fa(idO7) \geq Fa(idTh2)|1|1|Today\ 12:23:15|NodeC\}$$

Which describes the conclusion C_1:

$$C_1[idC1]:\{excessive\ packet\ loss|SLP|Fa(idO7) \geq Fa(idTh2)\}$$

9. The following symptom is reported to the Diagnosis layer:

$$Re_1[idRe1]:\{excessive\ packet\ loss|SLP|\text{idF1}|1|idO7, idTh2, Ru_1\}$$

All the temporal data related with objects O_2, thresholds T_{h2} and facts Fa is cleaned.

10. The ADB_3 with aggregated instances of O_1 and T_{h1} is requested to the Aggregation layer and processed. The SELFNET Analyzer builds the following facts from the information gathered by the network elements ($NodeA$, $NodeB$, $NodeC$, $NodeD$):

$$Fa(O)[idO09]:\{O_1 = 0.36|1|1|Today\ 12:24:15|NodeA\}$$
$$Fa(O)[idO10]:\{O_1 = 0.34|1|1|Today\ 12:24:15|NodeB\}$$
$$Fa(O)[idO11]:\{O_1 = 0.33|1|1|Today\ 12:24:15|NodeC\}$$
$$Fa(O)[idO12]:\{O_1 = 0.31|1|1|Today\ 12:24:15|NodeD\}$$

And by the data Aggregation:

$$Fa(Th)[idTh3]:\{T_{h1} = 0.77|1|1|Today\ 12:24:15|All\}$$

11. The Analyzer bypasses pattern recognition, prediction and adaptive thresholding.
12. The rule-based inference engine processes the discovered facts by applying the rule Ru_1. Given that the condition $Fa(O_1) \geq Fa(T_{h1})$ is not satisfied by any of the facts, conclusions related with ***SLP*** are not inferred.
13. All the temporal data related with objects O_1, thresholds T_{h1} and facts Fa is cleaned.

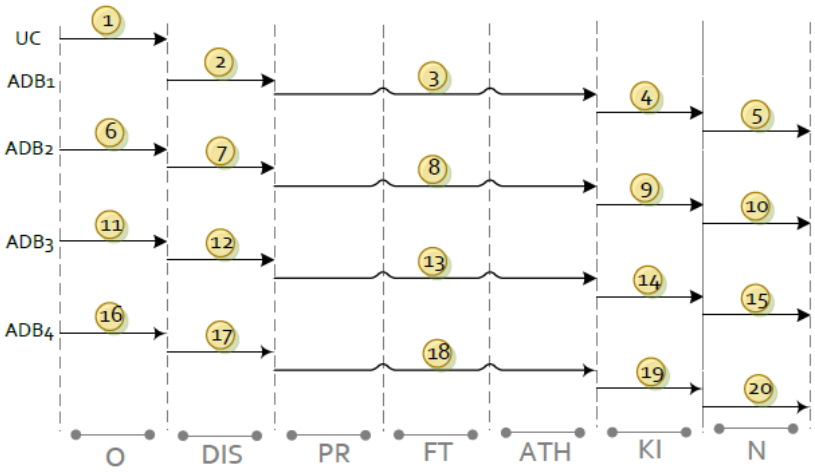

Figure 12: Example of Runtime in Self-packet Loss Prevention

7.2 UC 2: Quality of Service Analysis

7.2.1 Description

The illustrative use case to be managed $Self - QoSOverwatch$ (SQoS) report symptoms related with suspicious QoS decreasing, In particular, if a significantly decrement considering the latest observations is detected, a new fact related with relevant QoS variation is acquired. In this context, concurrency at pipelining is not applied, prediction and adaptive thresholding are considered, and it is assumed that the forecasting algorithm requires at least $n = 8$ observations for building the prediction model. Table 4 shows its onboarding descriptors according with the specification summarized in Table 2.

Table 4: $\boldsymbol{Self - QoSOverwatch}$ Specification

Item	Descriptor
Object	O_1:{$QoS\ decrement$|1|1|[0,1]}
Forecast	Ft_1:{$timeSeries\|O_1\|obs\|t+1$}
Adaptive Trehsold	AT_{h1}:{$maxQoS\ decrement\|timeSeries$|0.95|$Ft_1$}
Operator	Op_1:{$Equal$| = | 1 |$(Fa, O, Va) = (Fa, O, Va)$|equal}
Operator	Op_2:{LGT| ≥| 1 |$(Fa, O, Va) \geq (Fa, O, Va)$|left is GE}
Conclusion	C_1:{$Suspicious\ QoS\ variation\|SQoS\|Fa(O_1) \geq Fa(AT_{h1})$}
Rule	Ru_1:{$Fa(O_1) \geq Fa(AT_{h1}) \rightarrow Fa(C_1)$|1|$SC$}

7.2.2 Step-by-Step

The following illustrates and example of runtime in $Self - QoSOverwatch$, where different ADBs are loaded and analyzed according to the aforementioned indications. Fig. 13 displays every step in a sequence diagram, which are described step-by-step below:

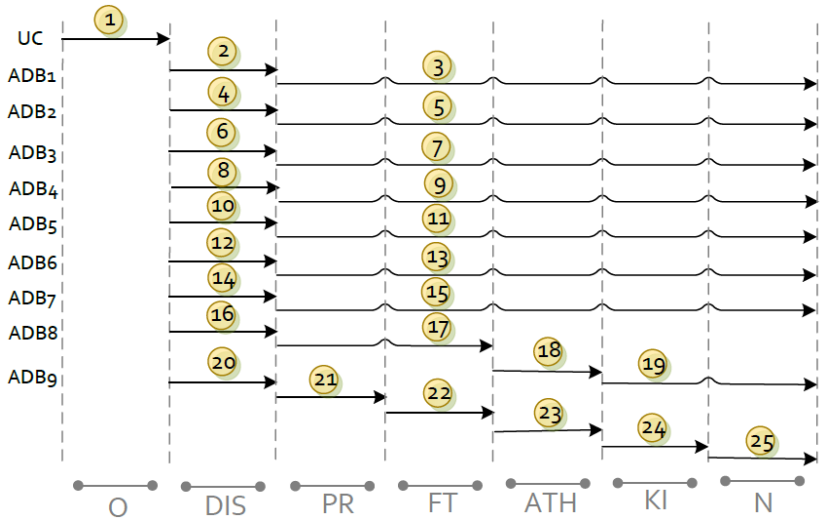

Figure 13: Example of Runtime in Self-QoSOverwatch

1. The descriptors of the $SQoS$ use case are loaded by the SELFNET Analyzer. Then, the memory for storing temporal containers of objects O_1, forecasts F_{t1}, adaptive tresholds AT_{h1} and facts Fa is allocated. Prediction capabilities on time series are required, so the data structures to support time series are initiated.

2. The ADB_1 with aggregated instances of O_1 is requested to the Aggregation layer and then it is processed. The SELFNET Analyzer built the following facts from the information gathered by the network elements ($NodeA$, $NodeB$, $NodeC$, $NodeD$):

$$Fa(O)[idO1]:\{O_1 = 0.60|1|1|Today\ 12:22:15|NodeA\}$$
$$Fa(O)[idO2]:\{O_1 = 0.65|1|1|Today\ 12:22:15|NodeB\}$$
$$Fa(O)[idO3]:\{O_1 = 0.61|1|1|Today\ 12:22:15|NodeC\}$$
$$Fa(O)[idO4]:\{O_1 = 0.62|1|1|Today\ 12:22:15|NodeD\}$$

3. Given that the Analyzer does not dispose of time series of $n = 8$ facts per sensor, prediction is not possible. Hence, adaptive thresholding is not performed. Because there are not facts related with adaptive thresholds, the rule Ru_1 where $Fa(O_1) \geq Fa(AT_{h1}) \rightarrow Fa(C_1)$ cannot be triggered. So conclusions related with symptoms are not notified to the diagnosis layer. On the other hand, given that the acquired facts are related with time series analysis (i.e. prediction and adaptive thresholding), they cannot be deleted before loading the following ADBs, but the rule-based inference engine is reinitiated.

4. The Analyzer performs the same actions (Step 2 and 3) from ADB_2 untilADB_7. Table 5 shows the facts built for these set of ADBs.

Table 5: Facts $\boldsymbol{ADB_2}$ **to** $\boldsymbol{ADB_7}$

ADB	Facts
$\boldsymbol{ADB_2}$	$Fa(O)[idO5]$:$\{O_1 = 0.63\|1\|1\|Today\ 12{:}23{:}15\|NodeA\}$ $Fa(O)[idO6]$:$\{O_1 = 0.64\|1\|1\|Today\ 12{:}23{:}15\|NodeB\}$ $Fa(O)[idO7]$:$\{O_1 = 0.65\|1\|1\|Today\ 12{:}23{:}15\|NodeC\}$ $Fa(O)[idO8]$:$\{O_1 = 0.66\|1\|1\|Today\ 12{:}23{:}15\|NodeD\}$
$\boldsymbol{ADB_3}$	$Fa(O)[idO9]$:$\{O_1 = 0.62\|1\|1\|Today\ 12{:}24{:}15\|NodeA\}$ $Fa(O)[idO10]$:$\{O_1 = 0.70\|1\|1\|Today\ 12{:}24{:}15\|NodeB\}$ $Fa(O)[idO11]$:$\{O_1 = 0.72\|1\|1\|Today\ 12{:}24{:}15\|NodeC\}$ $Fa(O)[idO12]$:$\{O_1 = 0.63\|1\|1\|Today\ 12{:}24{:}15\|NodeD\}$
$\boldsymbol{ADB_4}$	$Fa(O)[idO13]$:$\{O_1 = 0.60\|1\|1\|Today\ 12{:}25{:}15\|NodeA\}$ $Fa(O)[idO14]$:$\{O_1 = 0.72\|1\|1\|Today\ 12{:}25{:}15\|NodeB\}$ $Fa(O)[idO15]$:$\{O_1 = 0.73\|1\|1\|Today\ 12{:}25{:}15\|NodeC\}$ $Fa(O)[idO16]$:$\{O_1 = 0.65\|1\|1\|Today\ 12{:}25{:}15\|NodeD\}$
$\boldsymbol{ADB_5}$	$Fa(O)[idO17]$:$\{O_1 = 0.62\|1\|1\|Today\ 12{:}26{:}15\|NodeA\}$ $Fa(O)[idO18]$:$\{O_1 = 0.71\|1\|1\|Today\ 12{:}26{:}15\|NodeB\}$ $Fa(O)[idO19]$:$\{O_1 = 0.76\|1\|1\|Today\ 12{:}26{:}15\|NodeC\}$ $Fa(O)[idO20]$:$\{O_1 = 0.63\|1\|1\|Today\ 12{:}26{:}15\|NodeD$
$\boldsymbol{ADB_6}$	$Fa(O)[idO21]$:$\{O_1 = 0.63\|1\|1\|Today\ 12{:}27{:}15\|NodeA\}$ $Fa(O)[idO22]$:$\{O_1 = 0.70\|1\|1\|Today\ 12{:}27{:}15\|NodeB\}$ $Fa(O)[idO23]$:$\{O_1 = 0.71\|1\|1\|Today\ 12{:}27{:}15\|NodeC\}$ $Fa(O)[idO24]$:$\{O_1 = 0.60\|1\|1\|Today\ 12{:}27{:}15\|NodeD\}$
$\boldsymbol{ADB_7}$	$Fa(O)[idO25]$:$\{O_1 = 0.61\|1\|1\|Today\ 12{:}28{:}15\|NodeA\}$ $Fa(O)[idO26]$:$\{O_1 = 0.72\|1\|1\|Today\ 12{:}28{:}15\|NodeB\}$ $Fa(O)[idO27]$:$\{O_1 = 0.73\|1\|1\|Today\ 12{:}28{:}15\|NodeC\}$ $Fa(O)[idO28]$:$\{O_1 = 0.62\|1\|1\|Today\ 12{:}28{:}15\|NodeD\}$

5. The ADB_8 with aggregated instances of O_1 is requested to the Aggregation layer and then it is processed. The SELFNET Analyzer built the following facts from the information gathered by the network elements:

$$Fa(O)[idO29]:\{O_1 = 0.60|1|1|Today\ 12{:}29{:}15|NodeA\}$$
$$Fa(O)[idO30]:\{O_1 = 0.73|1|1|Today\ 12{:}29{:}15|NodeB\}$$
$$Fa(O)[idO31]:\{O_1 = 0.72|1|1|Today\ 12{:}29{:}15|NodeC\}$$
$$Fa(O)[idO32]:\{O_1 = 0.64|1|1|Today\ 12{:}29{:}15|NodeD\}$$

6. A this point, there are $n = 8$ facts per sensor in the time series to be predicted, so the forecasting method are able to estimate the next observation $(t + 1)$ as specified in the use case definition. The temporally stored data is summarized in Table 6.

Table 6: Summary of Information on Time Series at SQoS

Time	N	NodeA	NodeB	NodeC	NodeD
12:22:15	1	0.60	0.65	0.61	0.62
12:23:15	2	0.63	0.64	0.65	0.66
12:24:15	3	0.62	0.70	0.72	0.63
12:25:15	4	0.6	0.72	0.73	0.65
12:26:15	5	0.62	0.71	0.76	0.63
12:27:15	6	0.63	0.7	0.71	0.6
12:28:15	7	0.61	0.72	0.73	0.62
12:29:15	8	0.6	0.73	0.72	0.64
Forecast	n+1	0.61	0.72	0.72	0.63

The following facts related with prediction are acquired:

$$Fa(Ft)[idF1]:\{Ft_1 = 0.61|1|1|Today\ 12{:}29{:}15|NodeA\}$$
$$Fa(Ft)[idF2]:\{Ft_1 = 0.72|1|1|Today\ 12{:}29{:}15|NodeB\}$$
$$Fa(Ft)[idF3]:\{Ft_1 = 0.72|1|1|Today\ 12{:}29{:}15|NodeC\}$$
$$Fa(Ft)[idF4]\{Ft_1 = 0.63|1|1|Today\ 12{:}29{:}15|NodeD\}$$

7. The following facts related with the adaptive thresholds built from the predictions are acquired:

$$Fa(Ath)[idA1]:\{Ath_1 = 0.62|1|1|Today\ 12{:}29{:}15|NodeA\}$$
$$Fa(Ath)[idA2]:\{Ath_1 = 0.73|1|1|Today\ 12{:}29{:}15|NodeB\}$$
$$Fa(Ath)[idA3]:\{Ath_1 = 0.74|1|1|Today\ 12{:}29{:}15|NodeC\}$$
$$Fa(Ath)[idA4]\{Ath_1 = 0.64|1|1|Today\ 12{:}29{:}15|NodeD\}$$

8. The recent calculated thresholds are not applicable to the current observations, so the rule Ru_1 where $Fa(O_1) \geq Fa(AT_{h1}) \rightarrow Fa(C_1)$ cannot be triggered. Conclusions related with symptoms are not notified to the diagnosis layer.
   Note that for the observation i only the predictions and adaptive thresholds calculated at $0, \dots, i-1$ can be considered; stated in another way: predictions and adaptive thresholds calculated at i are only valid for the next $i+1$ observations, when it can be verified whether they have been fulfilled.

9. The ADB_9 with aggregated instances of O_1 is requested to the Aggregation layer and then it is processed. The SELFNET Analyzer built the following facts from the information gathered by the network elements ($NodeA, NodeB, NodeC, NodeD$):

$$Fa(O)[idO33]:\{O_1 = 0.60|1|1|Today\ 12{:}30{:}15|NodeA\}$$
$$Fa(O)[idO34]:\{O_1 = 0.82|1|1|Today\ 12{:}30{:}15|NodeB\}$$
$$Fa(O)[idO35]:\{O_1 = 0.62|1|1|Today\ 12{:}30{:}15|NodeC\}$$
$$Fa(O)[idO36]:\{O_1 = 0.60|1|1|Today\ 12{:}30{:}15|NodeD\}$$

10. Not pattern recognition actions are declared.

11. The following facts about predictions for the next observations are calculated:

$$Fa(Ft)[idF5]:\{Ft_1 = 0.60|1|1|Today\ 12{:}30{:}15|NodeA\}$$
$$Fa(Ft)[idF6]:\{Ft_1 = 0.75|1|1|Today\ 12{:}30{:}15|NodeB\}$$
$$Fa(Ft)[idF7]:\{Ft_1 = 0.72|1|1|Today\ 12{:}30{:}15|NodeC\}$$
$$Fa(Ft)[idF8]\{Ft_1 = 0.62|1|1|Today\ 12{:}30{:}15|NodeD\}$$

12. New facts related with adaptive thresholds are calculated:

$$Fa(Ath)[idA5]:\{Ath_1 = 0.61|1|1|Today\ 12{:}30{:}15|NodeA\}$$
$$Fa(Ath)[idA6]:\{Ath_1 = 0.76|1|1|Today\ 12{:}30{:}15|NodeB\}$$
$$Fa(Ath)[idA7]:\{Ath_1 = 0.73|1|1|Today\ 12{:}30{:}15|NodeC\}$$
$$Fa(Ath)[idA8]\{Ath_1 = 0.63|1|1|Today\ 12{:}30{:}15|NodeD\}$$

13. The rule-based inference engine processes the discovered facts by applying the rule Ru_1. Given that the condition $Fa(O_1) \geq Fa(AT_{h1})$ is satisfied for the data gathered by $NodeB$, The following fact related with $Self - QoSOverwatch$ is inferred.

$$Fa[\text{idF1}]:\{Fa(idO34) \geq Fa(idA2)|1|1|Today\ 12{:}30{:}15|NodeB\}$$

   Which describes the conclusion C_1:

$$C_1[idC1]:\{Suspicious\ QoS\ variation|SQoS|Fa(idO34) \geq Fa(idA2)\}$$

14. The following symptom is reported to the Diagnosis layer:

$$Re_1[idR1]:\{Suspicious\ QoS\ variation|SQoS|\text{idF1}|1|\text{idF1}, idA2, idO34, Ru_1\}$$

7.3 UC 3: Botnet Detection

7.3.1 Description

The use case called $Self-BotnetMitigation$ (SZombie) report symptoms related with Suspicious Command and Control (C&C) communications [33]. In this case concurrency at pipelining is not applied, pattern recognition actions are considered, but prediction and adaptive thresholding are not required. Note that the external repositories (Rep1, Rep2) provide collections of legitimate (Rep1) and malicious (Rep2) traffic pattern observations on SELFNET. When a new discovered patter seems much more significantly to the collection of malicious patterns than the legitimate, a new fact that indicates this suspicious feature is acquired. Table 7 shows its onboarding descriptors according with the specification summarized in Table 2.

Table 7: **Self − BotnetMitigation** Specification

Item	Descriptor
Object	O_1:{$pattern$|1|1|$hexadecimal$}
Object	O_2:{$simLegi$|1|1|{0..1}}
Object	O_3:{$simMal$|1|1|{0..1}}
Operator	Op_1:{$Equal$| = | 1 |$(Fa, O, Va) = (Fa, O, Va)$|equal}
Operator	Op_2:{LT| >| 1 |$(Fa, O, Va) > (Fa, O, Va)$|left is G}
Dataset	D_{legi}:{$legitimatePatern$|$O(pattern)$|$collection$|$Rep1$}
Dataset	D_{mal}:{$maliciousPatern$|$O(pattern)$|$collection$|$Rep2$}
Pattern Recognition	PR_1:{legMeasure|O_1|O_2|$anomaly$|D(D_{legi})}
Pattern recognition	PR_2:{$malMeasure$|O_1|O_3|$anomaly$|D(D_{mal})}
Conclusion	C_1:{$malicious\ communication$|$SZombie$|$Fa(O_2) < Fa(O_3)$}
Rule	Ru_1:{$Fa(O_2) < Fa(O_3) \rightarrow Fa(C_1)$|1|$SZombie$}

7.3.2 Step-by-Step

The following illustrates and example of runtime in $Self-BotnetMitigation$, where different ADBs are loaded and analyzed according to the aforementioned indications. Fig. 14 displays every step in a sequence diagram.

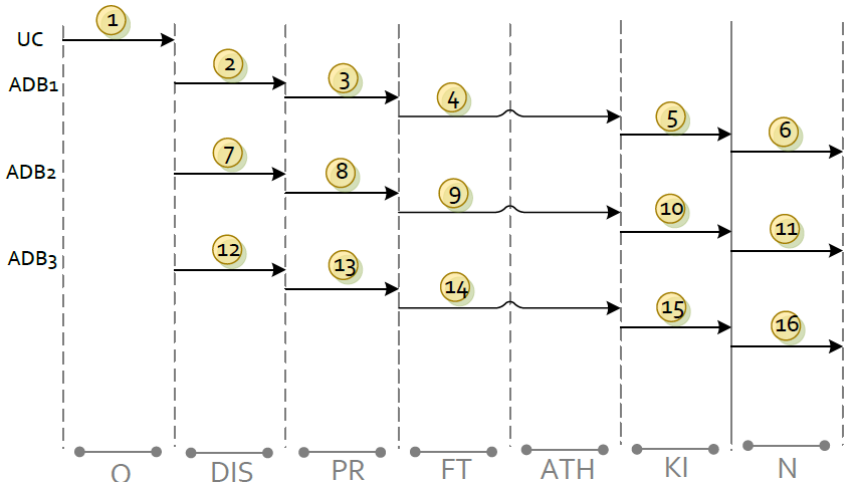

Figure 14: Example of Runtime in ***Self − BotnetMitigation***

1. The descriptors of the use case $Self - BotnetMitigation$ are loaded by the SELFNET Analyzer. Then, the memory for storing temporal containers of objects O_1, pattern recognition PR and facts Fa is allocated. The accessibility of the declared datasets D is verified.

2. The ADB_1 with aggregated instances of O_1 is requested to the Aggregation layer and then it is processed. The SELFNET Analyzer built the following facts from the information gathered by the network elements ($NodeA, NodeB, NodeC, NodeD$):

$$Fa(O_1)[idO1]{:}\{O_1 = FF217|1|1|Today\ 12{:}22{:}17|NodeA\}$$
$$Fa(O_1)[idO2]{:}\{O_1 = 00DE8|1|1|Today\ 12{:}22{:}17|NodeB\}$$
$$Fa(O_1)[idO3]{:}\{O_1 = F00FF|1|1|Today\ 12{:}22{:}17|NodeC\}$$
$$Fa(O_1)[idO4]{:}\{O_1 = A4F09|1|1|Today\ 12{:}22{:}17|NodeD\}$$

3. The two pattern recognition actions declared are executed. First, every $Fa(O_1)$ is correlated with the dataset D_{legi} looking for anomalies. The obtained anomaly scores O_2 allow acquiring the following facts:

$$Fa(PR)[idL1]{:}\{O_2 = 0.9|1|1|Today\ 12{:}22{:}17|NodeA\}$$
$$Fa(PR)[idL2]{:}\{O_2 = 0.9|1|1|Today\ 12{:}22{:}17|NodeB\}$$
$$Fa(PR)[idL3]{:}\{O_2 = 0.8|1|1|Today\ 12{:}22{:}17|NodeC\}$$
$$Fa(PR)[idL4]{:}\{O_2 = 0.9|1|1|Today\ 12{:}22{:}17|NodeD\}$$

   On the other hand, every $Fa(O_1)$ is correlated with the dataset D_{mali} looking for anomalies. The obtained anomaly scores O_3 allow acquiring the following facts:

$$Fa(PR)[idM1]{:}\{O_3 = 0.2|1|1|Today\ 12{:}22{:}17|NodeA\}$$
$$Fa(PR)[idM2]{:}\{O_3 = 0.1|1|1|Today\ 12{:}22{:}17|NodeB\}$$
$$Fa(PR)[idM3]{:}\{O_3 = 0.1|1|1|Today\ 12{:}22{:}17|NodeC\}$$
$$Fa(PR)[idM4]{:}\{O_3 = 0.1|1|1|Today\ 12{:}22{:}17|NodeD\}$$

4. There are not predictions or adaptive thresholding actions to execute.

5. The rule-based inference engine processes the discovered facts by applying the rule Ru_1. Given that the condition $Fa(O_2) < Fa(O_3)$ is not satisfied, conclusions related with $Self - BotnetMitigation$ are not inferred.

6. There are not symptoms to report. All the temporal facts are removed and the rule-based expert system is restarted.

7. The ADB_2 with aggregated instances of O_1 is requested to the Aggregation layer and then it is processed. The SELFNET Analyzer built the following facts from the information gathered by the network elements ($NodeA, NodeB, NodeC, NodeD$):

$$Fa(O_1)[idO5]{:}\{O_1 = F2217|1|1|Today\ 12{:}23{:}17|NodeA\}$$
$$Fa(O_1)[idO6]{:}\{O_1 = 012E8|1|1|Today\ 12{:}23{:}17|NodeB\}$$
$$Fa(O_1)[idO7]{:}\{O_1 = F0211|1|1|Today\ 12{:}23{:}17|NodeC\}$$
$$Fa(O_1)[idO8]{:}\{O_1 = A2F18|1|1|Today\ 12{:}23{:}17|NodeD\}$$

8. The two pattern recognition actions declared are executed. The following facts related with O_2 and O_3 are acquired:

$$Fa(PR)[idL5]{:}\{O_2 = 0.5|1|1|Today\ 12{:}23{:}17|NodeA\}$$
$$Fa(PR)[idL6]{:}\{O_2 = 0.8|1|1|Today\ 12{:}23{:}17|NodeB\}$$
$$Fa(PR)[idL7]{:}\{O_2 = 0.7|1|1|Today\ 12{:}23{:}17|NodeC\}$$
$$Fa(PR)[idL8]{:}\{O_2 = 0.9|1|1|Today\ 12{:}23{:}17|NodeD\}$$

$$Fa(PR)[idM5]:\{O_3 = 0.6|1|1|Today\ 12{:}23{:}17|NodeA\}$$
$$Fa(PR)[idM7]:\{O_3 = 0.2|1|1|Today\ 12{:}23{:}17|NodeB\}$$
$$Fa(PR)[idM8]:\{O_3 = 0.2|1|1|Today\ 12{:}23{:}17|NodeC\}$$
$$Fa(PR)[idM9]:\{O_3 = 0.3|1|1|Today\ 12{:}23{:}17|NodeD\}$$

9. There are not predictions or adaptive thresholding actions to execute.

10. The rule-based inference engine processes the discovered facts by applying the rule Ru_1. Given that the condition $Fa(O_2) < Fa(O_3)$ is satisfied for the data gathered by $NodeA$, conclusions related with $Self - BotnetMitigation$ are inferred. The following fact related with $Self - BotnetMitigation$ is included to the working memory.

$$Fa[\text{idF1}]:\{Fa(idL5) < Fa(idM5)|1|1|Today\ 12{:}23{:}17|NodeA\}$$

Which describes the conclusion C_1:

$$C_1[idC1]:\{malicious\ communication|SZombie|Fa(idL5) < Fa(idM5)\}$$

11. The following symptom is reported to the Diagnosis layer:

$$Re_1[idR1]:\{mmalicious\ communication|SZombie|\text{idF1}|1|\text{idF1}, idL5, idM5, Ru_1\}$$

12. The ADB_3 with aggregated instances of O_1 is requested to the Aggregation layer and then it is processed. The SELFNET Analyzer built the following facts from the information gathered by the network elements ($NodeA$, $NodeB$, $NodeC$, $NodeD$):

$$Fa(O_1)[idO9]:\{O_1 = F1110|1|1|Today\ 12{:}24{:}17|NodeA\}$$
$$Fa(O_1)[idOA]:\{O_1 = F2E80|1|1|Today\ 12{:}24{:}17|NodeB\}$$
$$Fa(O_1)[idOB]:\{O_1 = 11310|1|1|Today\ 12{:}24{:}17|NodeC\}$$
$$Fa(O_1)[idOC]:\{O_1 = AF42C|1|1|Today\ 12{:}24{:}17|NodeD\}$$

13. The two pattern recognition actions declared are executed. The following facts related with O_2 and O_3 are acquired:

$$Fa(PR)[idL9]:\{O_2 = 0.4|1|1|Today\ 12{:}24{:}17|NodeA\}$$
$$Fa(PR)[idLA]:\{O_2 = 0.3|1|1|Today\ 12{:}24{:}17|NodeB\}$$
$$Fa(PR)[idLB]:\{O_2 = 0.2|1|1|Today\ 12{:}24{:}17|NodeC\}$$
$$Fa(PR)[idLC]:\{O_2 = 0.9|1|1|Today\ 12{:}24{:}17|NodeD\}$$

$$Fa(PR)[idM9]:\{O_3 = 0.6|1|1|Today\ 12{:}24{:}17|NodeA\}$$
$$Fa(PR)[idMA]:\{O_3 = 0.5|1|1|Today\ 12{:}24{:}17|NodeB\}$$
$$Fa(PR)[idMB]:\{O_3 = 0.4|1|1|Today\ 12{:}24{:}17|NodeC\}$$
$$Fa(PR)[idMC]:\{O_3 = 0.2|1|1|Today\ 12{:}24{:}17|NodeD\}$$

14. There are not predictions or adaptive thresholding actions to execute.

15. The rule-based inference engine processes the discovered facts by applying the rule Ru_1. Given that the condition $Fa(O_2) < Fa(O_3)$ is satisfied for the data gathered by $NodeA$, $NodeB$ and $NodeC$, conclusions related with $Self - BotnetMitigation$ are inferred. The following facts related with $Self - BotnetMitigation$ are included to the working memory.

$$Fa[\text{idF2}]:\{Fa(idL9) < Fa(idM9)|1|1|Today\ 12{:}24{:}17|NodeA\}$$
$$Fa[\text{idF3}]:\{Fa(idLA) < Fa(idMA)|1|1|Today\ 12{:}24{:}17|NodeB\}$$
$$Fa[\text{idF4}]:\{Fa(idLB) < Fa(idMB)|1|1|Today\ 12{:}24{:}17|NodeC\}$$

Which describes the conclusion C_1:

$$C_1[idC2]: \{malicious\ communication|SZombie|Fa(idL9) < Fa(idM9)\}$$
$$C_1[idC3]: \{malicious\ communication|SZombie|Fa(idLA) < Fa(idMA)\}$$
$$C_1[idC4]: \{malicious\ communication|SZombie|Fa(idLB) < Fa(idMB)\}$$

16. The following symptoms are reported to the Diagnosis layer:

$$Re_2[idR2]:\{malicious\ communication|SZombie|\text{idF2}|1|\text{idF2}, idL9, idM9, Ru_1\}$$
$$Re_3[idR3]:\{malicious\ communication|SZombie|\text{idF3}|1|\text{idF3}, idLA, idMA, Ru_1\}$$
$$Re_4[idR4]:\{malicious\ communication|SZombie|\text{idF4}|1|\text{idF4}, idLB, idMB, Ru_1\}$$

8 Conclusions

This paper described the key elements of the SELFNET Analyzer Orchestrator, including the initial assumptions, design principles, workflows, sets of actions, execution strategies, and several examples of their application. These were properly deployed in the SELFNET project, where their effectiveness, configurability and extensibility were verified at different use cases (in particular, when applied to self-protection, self-optimization and self-healing capabilities). But even though our approach has proved to meet its design objectives, throughout the document has remained several aspects without covering, which depend directly on the implementation. For instance, some alternatives have been described for their optimization, but at the moment, only the basic task sequencing approach was being implemented on real uses cases. It brings support to the most basic requirements of the system, and allows the verification of the analyzed communication channels. But it is clear that there are many other ways to exploit parallelism at thread level, and therefore, to take more advantage of the analytic pipelining. On the other hand, the proposed scheme forces the most complex sets of actions to be executed in a certain way: pattern recognition, prediction, adaptive threshold construction and knowledge inference. But it is possible that future uses cases demand variations in their order; for example, that pattern recognition is being carried considering facts related with prediction and adaptive thresholds, or that once the knowledge inference tasks are completed, an additional step of predictions is required. At the moment, easy modifications on the SELFNET use case descriptors are able to overcome this inconvenience, but it is obvious that deepen in this problem is one of the main tasks of future work. Another aspect of interest is identifying quality indicators related with the granularity of the information contained in the ADBs. From them it is possible to improve the effectiveness of the analytic actions.

References

[1] ENISA, "ENISA Threat Landscape 2015" (RTL 2015), Available at: https://www.enisa.europa.eu/publications/etl2015

[2] International Organization for Standardization and the International Electrotechnical Commission. "ISO/IEC 27002: Information Technology, Security Techniques, Code of Practice for Information Security Management". 2005. Available online: http://www.iso.org/iso/catalogue_detail?csnumber=54533 (accessed on 30 January 2017).

[3] National Institute of Standards and Technology. "NIST-SP800 Series Special Publications on Computer Security". Available online: http://csrc.nist.gov/publications/PubsSPs.html#SP800 (accessed on 30 January 2017).

[4] Forum of Incident Response and Security Teams. "CVSS: Common Vulnerability Scoring System". Available online: https://www.first.org/cvss/specification-document (accessed on 30 January 2017).

[5] J. Webb, A. Ahmad , S.B. Maynard, G. Shanks, P. Popovski, "A Situation Awareness Model for Information Security Risk Management". Computers & Security Vol. 44, pp. 1-15, April 2014.

[6] N.R. Endsley, “Design and Evaluation for Situation Awareness Enhancement”. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Anaheim, CA, USA, 1988; Vol. 32(2); pp. 97-101.

[7] Y.B. Leau, A. Ahmad, S. Manickam, “Network Security Situation Prediction: A Review and Discussion”. In Proceedings of the 4th International Conference on Soft Computing, Intelligent Systems, and Information Technology (ICSIIT), Bali, Indonesia, pp. 424-435, March 2015.

[8] A. Osseiran, F.Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, M.A. Uusitalo, B. Timus, M. Fallgren, "Scenarios for 5G mobile and wireless communications: the vision of the METIS project", IEEE Communications Magazine, Vol. 52, Issue 5, pp. 26-35, May 2014.

[9] R. Mijumbi, J. Serrat, J.L. Gorricho, N. Bouten, F. De Turck, R. Boutaba, "Network Function Virtualization: State-of-the-Art and Research Challenges", IEEE Communications Surveys & Tutorials, Vol. 18, Issue 1, pp. 236-262, Firstquarter 2016.

[10] W. Xia, Y. Wen, C.H. Foh, D. Niyato, H. Xie, "A Survey on Software-Defined Networking", IEEE Communications Surveys & Tutorials, Vol. 17, Issue 1, pp. 27-51, Firstquarter 2015.

[11] J. Qiao, X.S. Shen, J.W. Mark, Q. Shen, Y. He, L. Lei, "Enabling device-to-device communications in millimeter-wave 5G cellular networks", IEEE Communications Magazine, Vol. 53, Issue 1, pp. 209-2015, January 2015.

[12] N. Baldo, L. Giupponi, J. Mangues-Bafalluy. “Big Data Empowered Self Organized Networks”. In Proceedings of 20th European Wireless Conference, Barcelona, Spain, pp. 1-8, May 2014.

[13] 5G Infrastructure Public Private Partnership - 5G PPP. Available online: https://5g-ppp.eu

[14] L.I. Barona López, A.L. Valdivieso Caraguay, M.A. Sotelo Monge, L.J. García Villalba, "Key Technologies in the Context of Future Networks: Operational and Management Requirements", Future Internet, Vol. 0(1), No. 1, December 2016.

[15] MCN Project. Funded under: FP7-ICT. Project Reference: 318109, Funded under: FP7-ICT. Available online: http://www.mobile-cloud-networking.eu/site/ (accessed on 30 January 2017).

[16] EU T-NOVA Project. Network Functions as-a-Service over Virtualised Infrastructures. Project Reference: 619520. Funded under: FP7-ICT. Available online: http://www.t-nova.eu/ (accessed on 30 January 2017).

[17] EU UNIFY Project. Unifying Cloud and Carrier Networks. Project Reference: 619609. Funded under: FP7-ICT. Available online: http://www.fp7-unify.eu/ (accessed on 30 January 2017).

[18] EU CROWD Project. Connectivity Management for EneRgy Optimised Wireless Dense Networks. Project Reference: 318115. Funded under: FP7-ICT. Available online: http://www.ict-crowd.eu/ (accessed on 30 January 2017).

[19] 5G-NORMA Project. 5G NOvel Radio Multiservice Adaptive Network Architecture. Project Reference: 671584. Funded under: H2020-ICT-2014-2. Available online: https://5gnorma.5g-ppp.eu/ (accessed on 30 January 2017).

[20] CHARISMA Project. Converged Heterogeneous Advanced 5G Cloud-RAN Architecture for Intelligent and Secure Media Access. Project Reference: 671704. Funded under: H2020-ICT-2014-2. Available online: http://www.charisma5g.eu// (accessed on 30 January 2017).

[21] SELFNET Project. Framework for Self-Organized Network Management in Virtualized and Software Defined Networks. Project reference: 671672. Funded under: H2020-ICT-2014-2. Available online: https://SELFNET-5g.eu/ (accessed on 30 January 2017).

[22] Xu, L.; Assem, H.; Yahia, I.G.B.; Buda, T.S.; Martin, A.; Gallico, D.; Biancani, M.; Pastor, A.; Aranda, P.A.; Smirnov, M.; et al. CogNet: A Network Management Architecture Featuring Cognitive Capabilities. IEEE Netw. Commun. (EuCNC) 2016, 2016, 325–329.

[23] L.I. Barona López, A.L. Valdivieso Caraguay, J. Maestre Vidal, M.A. Sotelo Monge, L.J. García Villalba, "Towards Incidence Management in 5G Based on Situational Awareness", Future Internet, Vol. 9(1), No.3, January 2017.

[24] L.I. Barona López, J. Maestre Vidal, L.J. García Villalba: "An Approach to Data Analysis in 5G Networks". MDPI Entropy, Vol. 19(2), pp. 1-23, February 2017.

[25] N. Bassiliades, I. Vlahavas, "Data & Knowledge Engineering", Vol. 24, Issue 2, pp. 117-155, November 1997.

[26] S. Guillaume, B. Charnomordic, "Fuzzy inference systems: An integrated modeling environment for collaboration between expert knowledge and data using FisPro", Expert Systems with Applications, Vol. 39, Issue 10, pp. 8744-8755, August 2012.

[27] A.D Lunardhi, K.M. Passino, "Verification of qualitative properties of rule-based expert systems". Applied Artificial Intelligence an International Journal, 9(6), 587-621, 1995.

[28] Y.W. Wang, E.N. Hanson, "A performance comparison of the Rete and TREAT algorithms for testing database rule conditions", in Proc. 8th International Conference on Data Engineering, Tempe, AZ, USA, pp. 88-97, February 1992.

[29] A. Finkel, S.P. Iyer, G. Sutre, "Well-abstracted transition systems: application to FIFO automata", Information and Computation, Vol. 181, Issue 1, pp. 1-31, February 2003.

[30] H. Zou, Y. Yu, W. Tang, H.W.M. Chen, "FlexAnalytics: A Flexible Data Analytics Framework for Big Data Applications with I/O Performance Improvement", Big Data Research, Vol. 1, pp. 4-13, August 2014.

[31] M.I.Gordon, W. Thies, S. Amarasinghe, "Exploiting coarse-grained task, data, and pipeline parallelism in stream programs", In proc. 12th International Conference on Architectural Support for Programming Languages and Operating Systems, San Jose, CA, USA, pp. 151-162, October 2006.

[32] J. Su, C. Xu, S.C. Chenung, W. Xi, Y. Jiang, C. Cao, X. Ma, J. Lu, "Hybrid CPU–GPU constraint checking: Towards efficient context consistency", Information and Software Technology, Vol. 74, pp. 230-242, June 2016.

[33] T.S. Wang, H.T. Lin, W.T. Cheng, C.Y. Chen, "DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis", Computers & Security, Vol. 64, pp. 1-15, January 2017.

[34] 5G-Ensure Project. Enablers for Network and System Security and Resilience. Project Reference: 671562. Funded under: H2020-ICT-2014-2. Available online: http://www.5gensure.eu/ (accessed on 30 January 2017).

[35] SONATA Project. Service Programing and Orchestration for Virtualized Software Networks. Project. Reference: 671517. Funded under: H2020-ICT-2014-2. Available online: http://www.sonata-nfv.eu/ (accessed on 30 January 2017).

[36] 5G-NOW Project. 5th Generation Non-Orthogonal Waveforms for Asynchronous Signalling Project. Reference 318555. Funded under: FP7-ICT. Available online: http://www.5gnow.eu/ (accessed on 30 January 2017).

[37] METIS-II Project. Mobile and Wireless Communications Enablers for Twenty-Twenty (2020) Information Society-II. Project Reference: 671680. Funded under: H2020-ICT-2014-2. Available online: https://5g-ppp.eu/metis-ii/ (accessed on 30 January 2017).

[38] Neves, P.; Calé, R.; Costa, M.R.; Parada, C.; Parreira, B.; Alcaraz-Calero, J.; Wang, Q.; Nightingale, J.; Chirivella-Perez, E.; Jiang, W.; et al. The SELFNET Approach for Autonomic Management in an NFV/SDN Networking Paradigm. Int. J. Distrib. Sens. Net. 2016, 2016, 1–17.