Department of Computer Architecture
Technical University of Catalonia (UPC)

Doctoral Thesis

---

# An Ontology-Based Approach Toward the Configuration of Heterogeneous Network Devices

---

***Author:***
Anny Martinez

***Advisor:***
Dr. Marcelo Yannuzzi
***Co-advisor:***
Dr. Xavi Masip-Bruin

*A THESIS PRESENTED TO THE TECHNICAL UNIVERSITY OF CATALONIA IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF*

*DOCTOR IN COMPUTER SCIENCE*

Barcelona, April 2015

*-"Intelligence" is the art of good guessing.-*
H. Barlow

TECHNICAL UNIVERSITY OF CATALONIA (UPC)

# *Abstract*

Department of Computer Architecture

Doctor in Computer Science

**An Ontology-Based Approach Toward the Configuration of Heterogeneous Network Devices**

by Anny Martinez

**Keywords:** *Networking*; *Semantics*; *Network Management*; *Future Internet*; *Ontology-Based Information Extraction*; *Device Configuration*; *Command Line Interface*; *LISP*; *Addressing Scheme.*

Despite the numerous efforts of standardization, semantic issues remain in effect in many subfields of networking. The inability to exchange data unambiguously between information systems and human resources is an issue that hinders technology implementation, semantic interoperability, service deployment, network management, technology migration, among many others. In this thesis, we will approach the semantic issues in two critical subfields of networking, namely, network configuration management and network addressing architectures. The fact that makes the study in these areas rather appealing is that in both scenarios semantic issues have been around from the very early days of networking. However, as networks continue to grow in size and complexity current practices are becoming neither scalable nor practical.

One of the most complex and essential tasks in network management is the configuration of network devices. The lack of comprehensive and standard means for modifying and controlling the configuration of network elements has led to the continuous and extended use of proprietary Command Line Interfaces (CLIs). Unfortunately, CLIs are generally both, device and vendor-specific. In the context of heterogeneous network infrastructures—i.e., networks typically composed of multiple devices from different vendors—the use of several CLIs raises serious Operation, Administration and Management (OAM) issues. Accordingly, network administrators are forced to gain specialized expertise and to continuously keep knowledge and skills up to date as new features, system upgrades or technologies appear. Overall, the utilization of proprietary mechanisms allows neither sharing knowledge consistently between vendors' domains nor reusing configurations to achieve full automation of network configuration tasks—which are typically required in autonomic management. Due to this heterogeneity, CLIs

typically provide a help feature which is in turn an useful source of knowledge to enable semantic interpretation of a vendor's configuration space. The large amount of information a network administrator must learn and manage makes Information Extraction (IE) and other forms of natural language analysis of the Artificial Intelligence (AI) field key enablers for the network device configuration space. This thesis presents the design and implementation specification of the first Ontology-Based Information Extraction (OBIE) System from the CLI of network devices for the automation and abstraction of device configurations.

Moreover, the so-called semantic overload of IP addresses—wherein addresses are both identifiers and locators of a node at the same time—is one of the main constraints over mobility of network hosts, multi-homing and scalability of the routing system. In light of this, numerous approaches have emerged in an effort to decouple the semantics of the network addressing scheme. In this thesis, we approach this issue from two perspectives, namely, a non-disruptive (i.e., evolutionary) solution to the current Internet and a clean-slate approach for Future Internet. In the first scenario, we analyze the Locator/Identifier Separation Protocol (LISP) as it is currently one of the strongest solutions to the semantic overload issue. However, its adoption is hindered by existing problems in the proposed mapping systems. Herein, we propose the LISP Redundancy Protocol (LRP) aimed to complement the LISP framework and strengthen feasibility of deployment, while at the same time, minimize mapping table size, latency time and maximize reachability in the network. In the second scenario, we explore TARIFA a Next Generation Internet architecture and introduce a novel service-centric addressing scheme which aims to overcome the issues related to routing and semantic overload of IP addresses.

# *Acknowledgements*

This thesis would not have been possible without the inspiration and support of many people. I would like to begin by expressing my sincere gratitude to my Ph.D. advisors Dr. Marcelo Yannuzzi and Dr. Xavier Masip-Bruin for their valuable guidance, support and above all for becoming referents of professionalism in this journey. Their dedication, invaluable discussions and timely advices certainly made this work possible.

Furthermore, I would like to extend my appreciation to Rafael Mulero for his commitment and collaboration in the implementation and demonstration of the work herein developed. Many thanks to my colleagues at CRAAX and NetIT with whom I had the pleasure of working over these past years. Moreover, I would like to thank the administrative staff at UPC for their support during the course of my thesis.

Additionally, I would like to acknowledge Dr. Jorge López-De Vergara for his valuable insights on my work and support in joint papers.

Sincere thanks are extended to the members of my thesis committee Dr. Marilet De Andrade and Dr. Mario Nemirovsky, as well as internal reviewers, for their valuable time and dedication to the evaluation of my work.

In the course of my thesis several organizations have been key supporters of my work. I would like to thank the TARIFA Project (i2CAT) and the European Commission through the ONE Project in the FP7 Program under contract number INFSO-ICT-258300. I would also like to acknowledge the Spanish Ministry of Science and Innovation under contracts TEC2009-07041 and TEC2012-34682, the Catalan Government under contract 2009 SGR1508 and the Technical University of Catalonia (UPC).

Finally, a very special thanks to my parents, my sister and most importantly, to Lorenzo for all their inspiring words, for believing in me even in the most difficult moments, and for always encouraging me to go a step further. Words cannot ever express my gratitude for their support and trust.

Anny Martínez, March 2015

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| **AI** | **A**rtificial **I**ntelligence |
| **API** | **A**pplication **P**rogramming **I**nterface |
| **AS** | **A**utonomous **S**ystem |
| **BDM** | **B**alanced **D**istance **M**etric |
| **BRPC** | **B**ackward **R**ecursive **P**ath **C**omputation |
| **BFS** | **B**readth **F**irst **S**earch |
| **CAPEX** | **CAP**ital **EX**penditures |
| **CC** | **C**ontinuity **C**heck |
| **CCV** | **C**ommon **C**ommunication **V**ehicle |
| **CLI** | **C**ommand **L**ine **I**nterface |
| **CMOS** | **C**omplementary **M**etal-**O**xide **S**emiconductor |
| **CV** | **C**onnectivity **V**erification |
| **DHT** | **D**istributed **H**ash **T**able |
| **DNS** | **D**omain **N**ame **S**ystem |
| **DWDM** | **D**ense **W**avelength-**D**ivision **M**ultiplexing |
| **ECMP** | **E**qual **C**ost **M**ulti-**P**ath |
| **EICTA** | **E**uropean **I**nformation & **C**communications **T**echnology Industry **A**ssociation |
| **EID** | **E**ndpoint **ID**entifier |
| **ETR** | **E**gress **T**unnel Router |
| **FI** | **F**uture **I**nternet |
| **GATE** | **G**eneral **A**rquitecture for **T**ext **E**ngineering |
| **GMPLS** | **G**eneralized **M**ulti-**P**rotocol **L**abel **S**witching |
| **HSRP** | **H**ot **S**tandby **R**outing **P**rotocol |
| **ICT** | **I**nformation and **C**ommunications **T**echnology |
| **IE** | **I**nformation **E**xtraction |

| | |
|---|---|
| **IETF** | **I**nternet **E**ngineering **T**ask **F**orce |
| **IoT** | **I**nternet **o**f **T**hings |
| **IP** | **I**nternet **P**rotocol |
| **IP-NMS** | **IP** **N**etwork **M**anagement **S**ystem |
| **IPoDWDM** | **IP** over **DWDM** |
| **ISP** | **I**nternet **S**ervice **P**rovider |
| **ITR** | **I**ngress **T**unnel **R**outer |
| **ITU** | **I**nternational **T**elecommunication **U**nion |
| **LCS** | **L**east **C**ommon **S**ubsumer |
| **LISP** | **L**ocator/**I**dentifier **S**eparation **P**rotocol |
| **LMP** | **L**ink **M**anagement **P**rotocol |
| **LRP** | **L**ISP **R**edundancy **P**rotocol |
| **LSP** | **L**abel **S**witched **P**ath |
| **MIB** | **M**anagement **I**nformation **B**ase |
| **MLI** | **M**ulti-**L**ayer **I**nteroperability |
| **MLNM** | **M**ulti-**L**ayer **N**etwork **M**anagement |
| **MPLS** | **M**ulti-**P**rotocol **L**abel **S**witching |
| **MPLS-TP** | **M**ulti-**P**rotocol **L**abel **S**witching **T**ransport **P**rofile |
| **MPTCP** | **M**ulti-**P**ath **T**ransport **C**ontrol **P**rotocol |
| **MTNM** | **M**ulti-**T**echnology **N**etwork **M**anagement |
| **MTOSI** | **M**ulti-**T**echnology **O**peration **S**ystems **I**nterface |
| **MVI** | **M**ulti-**V**endor **I**nteroperability |
| **NFV** | **N**etwork **F**unctions **V**irtualization |
| **NLP** | **N**atural **L**anguage **P**rocessing |
| **NMS** | **N**etwork **M**anagement **S**ystem |
| **OAM&P** | **O**peration, **A**dministration, **M**aintenance and **P**rovisioning |
| **OBIE** | **O**ntology-**B**ased **I**nformation **E**xtraction |
| **OEO** | **O**ptical-**E**lectrical-**O**ptical |
| **OF-CONFIG** | **O**pen**F**low **CONFIG**uration Protocol |
| **OIF** | **O**ptical **I**nternetworking **F**orum |
| **OPEX** | **OP**erational **EX**penditures |
| **OS** | **O**perating **S**ystem |
| **OSI** | **O**pen **S**ystems **I**nterface |

| | |
|---|---|
| **OSS** | **O**perations **S**upport **S**ystem |
| **OTN** | **O**ptical **T**ransport **N**etwork |
| **OVS** | **O**pen **VS**witch |
| **OWL** | **W**eb **O**ntology **L**anguage |
| **OXC** | **O**ptical **C**ross-**C**onnect |
| **PA** | **P**rovider-**A**ggregatable |
| **PCC** | **P**ath **C**omputation **C**lient |
| **PCE** | **P**ath **C**omputation **E**lement |
| **PCEP** | **P**ath **C**omputation **E**lement **P**rotocol |
| **PHP** | **P**enultimate **H**op **P**opping |
| **PI** | **P**rovider-**I**ndependent |
| **POS** | **P**art **O**f **S**peach |
| **QoS** | **Q**uality **o**f **S**ervice |
| **RDF** | **R**esource **D**escription **F**ramework |
| **RFC** | **R**equest **F**or **C**omments |
| **RLOC** | **R**outing **LOC**ator |
| **ROADM** | **R**econfigurable **O**ptical **A**dd **D**rop **M**ultiplexers |
| **SDH** | **S**ynchronous **D**igital **H**ierarchy |
| **SDN** | **S**oftware **D**efined **N**etworking |
| **SLA** | **S**ervice **L**evel **A**greement |
| **SNMP** | **S**imple **N**etwork **M**anagement **P**rotocol |
| **SONET** | **S**ynchronous **O**ptical **N**etworking |
| **SPF** | **S**hortest **P**ath **F**irst |
| **SWRL** | **S**emantic **W**eb **R**ule **L**anguage |
| **TE** | **T**raffic **E**ngineering |
| **TED** | **T**raffic **E**ngineering **D**atabase |
| **TMF** | **T**ele **M**anagement **F**orum |
| **T-MPLS** | **T**ransport **M**ulti-**P**rotocol **L**abel **S**witching |
| **T-NMS** | **T**ransport **N**etwork **M**anagement **S**ystem |
| **UNI** | **U**ser-**N**etwork **I**nterface |
| **URL** | **U**niform **R**esource **L**ocator |
| **UUID** | **U**niversally **U**nique **ID**entifier |
| **VNT** | **V**irtual **N**etwork **T**opology |

| | |
|---|---|
| **VNTM** | **V**irtual **N**etwork **T**opology **M**anager |
| **WSON** | **W**avelength **S**witched **O**ptical **N**etworks |
| **XML** | E**X**tensible **M**arkup **L**anguage |
| **XSD** | **XML S**chema |

*To my parents and Lorenzo...*

# Part I

# INTRODUCTION

# Chapter 1

# Summary and Road Map

*This introductory chapter provides a thorough description of the motivations and main contributions of this thesis. The rest of the chapter enumerates supporting publications to journals and conferences and concludes with an overview of the structure of this manuscript.*

## 1.1 Motivations

The exponential growth of both Internet traffic and digital information (big data) give no respite to the telecommunications industry and is visibly shortening the life-cycle of the technologies used for core networking. To cope with the traffic demand, the industry has primarily focused on the evolution of the data and control planes, and has rapidly made progress in both subjects. Accordingly, a variety of protocols and technologies have emerged at a very rapid pace, bringing along a complex set of new terminologies and conceptualizations, which in turn raise serious semantic issues.

The semantic problems in the field of networking are related to the inability to understand and unambiguously interpret the meaning of domain concepts and protocol units for different purposes, for instance, implementation, management, information modeling, service deployment, etc. Overall, these problems derive either from: *(i)* the lack of absolute standards for the performance of network functions—which in turn lead to the use of proprietary solutions that typically raise serious interoperability issues in the context of heterogeneous environments; *(ii)* the lack of formal means for the specification of network protocols—a fact which unquestionably leads to ambiguity and incompleteness of conceptualizations; *(iii)* the intrinsic nature of an ever-changing domain flood of terminologies—wherein dissimilar concepts are often used interchangeably to refer

to the same semantics; or *(iv)* the semantic overload of domain entities (e.g., Internet Protocol - IP address semantic overload [4]).

The semantic issues in the field of networking have been around for long time. Consider for instance, the challenges faced for the implementation and conformance testing due to the loose specification of networking protocols in public Request for Comments' (RFCs) [5]; or the multiplicity of network management information languages and their unique level of semantic expressiveness [6]. In this thesis, we will approach the semantic problems in two critical areas of networking, namely, the configuration management of network devices and the semantic overload of the current IP addressing architecture—an issue with serious implications in the scalability of the global routing system.

### 1.1.1 Semantic Issues in Configuration Management

One of the most complex and essential tasks in network management is network device configuration. Overall, it encompasses a large number of functions, which include, setting routing protocols, security filters, interface parameters, forwarding rules, Quality of Service (QoS) policies, etc. Despite numerous efforts, network administrators continue to rely on Command-Line Interfaces (CLIs) as the preferred means for configuring their network devices [7]. This is typically the case of administrators operating in Internet Service Providers (ISPs), data centers, corporate networks, public administrations, etc. Due to the lack of standards, administrators must deal with the complexities associated with this practice, since CLIs are generally both, device and vendor-specific, meaning that, commands syntax and semantics are specific to each configuration environment. This is further exacerbated by the heterogeneity of today's network infrastructures, as networks are typically composed of devices from different vendors (i.e., multi-vendor networks). Accordingly, network administrators are forced to gain specialized expertise for a wide range of devices, and to continuously keep knowledge and skills up to date as new features, operating system upgrades, or technologies appear in the routing market. In light of all this, device misconfiguration is a common event, which often leads to serious service disruptions [8].

From reviewing the literature, it becomes evident that, over the last few years, academic and industrial communities have devoted considerable efforts to overcome the inherent complexities of the so-mentioned CLIs or the cumbersome and rarely used configuration means provided by the Simple Network Management Protocol (SNMP) [9]. Clearly, the need for standards has always been among the best interests of the Internet community, since they are essential to achieve interoperability at a global level. In light of this, the NETCONF protocol [10] emerged as an attempt to standardize device configurations.

Unfortunately, NETCONF itself has not gained momentum yet, so it remains to be seen if it will finally become the protocol of choice [11]. In this regard, industry sources state that nearly 95% of network devices are still configured through proprietary Command-Line Interfaces (CLIs) [12]. The reason for this is the lack of comprehensive and widely accepted data models. This gap was recently filled by YANG [13], a candidate language for developing standardized data models for NETCONF. Still, four years after its standardization, only few YANG data models have found broad acceptance [14]. Moreover, several industrial initiatives have also attempted to provide uniform configuration means, by developing and maintaining dedicated software agents. Unfortunately, this approach demands serious development efforts, which are neither scalable nor easy to maintain under the dynamics of current networking environments.

In light of this, there is an imminent need to explore other fields that can help pave the way toward seamless network device configuration. We strongly believe that a solution which can assist in the configuration of network devices, is less a matter of developing new ways of managing the network or adding new protocols that boost complexity, but more of adapting well-known techniques that have proven to be absolute trends in the configuration field from the earliest days of networking. For this reason, legacy Command-Line Interfaces can help bridge the configuration gap. One of the most relevant features of CLIs is the availability of textual resources for supporting and guiding network administrators on its use. The exploitation of knowledge from natural language has been the focus of several research initiatives in numerous domains wherein text-based resources are largely available, but—to the best of our knowledge—researchers in the networking field have not explored this path yet. Accordingly, there is still a long research path to follow in order to significantly benefit from the field of Information Extraction, ontologies and semantic technologies for networking.

### 1.1.2 Semantic Issues in Addressing Schemes

In the past years, early forms of ubiquitous communication have arisen and become more evident as society expectations toward technology increases. These facts seem to prove that current Internet will naturally evolve to an Internet of Things (IoT) as a new dynamic communication scheme where objects, services, spaces and even people may be given a unique number, almost avoiding barriers for recognizing, locating, addressing, reaching, controlling and enjoying almost anything via the Internet, through a mix of heterogeneous wired and wireless network infrastructures.

However, the current Internet semantic overload of addresses—where addresses are simultaneously referred to identifiers and locators of a node—is an important constraint

over mobility and scalability concerns. In such a scheme, whenever a host changes of network provider its IP address changes as well—changing not only the network providing host access, but also the host identifier. For the case of upper layer applications wherein IP addresses are fixed for a given host (i.e., hard-coded) this represents a severe mobility constraint. In a world where a huge volume of objects can be uniquely identified and where objects capacity of mobility increases over time, decoupling of naming and location seems to be one of the first steps toward the evolution to an IoT. Several factors, such as the rise of multihoming sites, semantic overloading of IP addresses, among others, affect the scalability of the global Internet Routing Table, thus, fueling its size and dynamics. In an effort to solve these issues, academics have followed one of two research lines.

On the one hand, a first research line pursuits evolutionary solutions for the current Internet. In this line, several proposals aimed to decouple the semantics of the addressing scheme have emerged, namely, SHIM6 [15], Six/One Router [16], HIP [17], Multi-Path TCP [18], GSE [19], and LISP [20]. The latter is one of the strongest solutions so far as already considered by the Internet Engineering Task Force (IETF). The LISP concept is based on the notion of decoupling the dual semantics of host identification (Endpoint identifier, EID) and location (Routing locator, RLOC). The scaling benefits arise because EID addresses are not routable through the Internet—i.e., EIDs are only locally distributed. Accordingly, efficient aggregation of the RLOC address space is achieved, significantly reducing the overall routing load throughout the network. Authors in [21] show that the size of the global routing table can be reduced by roughly two orders of magnitude with LISP. While from a deployment perspective LISP is a non-disruptive approach, the existing control plane proposals have some major challenges. These challenges lie in the fact that because EIDs are not globally routable through the Internet, a mapping system is required between EIDs and RLOCs. LISP does not specify a mandatory mapping system, and as a consequence, different proposals can be found in the recent literature, such as ALT [22], NERD [23] or Map Server [24]. Most recently, authors in [25] introduce a new LISP control plane which is based on the idea of retrieving EID-to-RLOC mappings within the Domain Name System (DNS) Resolution time. Despite the fact that the proposed control plane presents an improvement—with respect to existing state-of-the-art solutions—over three relevant aspects, namely, *(i)* the first packet drop problem (i.e., whenever an ITR does not have a mapping for an EID-prefix); *(ii)* potential increase in the latency to start a communication due to the mapping resolution; and *(iii)* the ITR being used as the local ETR for the packets sent from destination to source, in order to avoid a two-way mapping resolution. A major shortcoming is that the mappings between EIDs and RLOCs are replicated in all of the edge routers within the same AS. Though this ensures improved reachability,

unfortunately, it may bring scalability problems since each router must store mapping information that rarely needs to use, thus increasing the latency time to find a mapping. This new issue directly affects the memory component within the router, which is currently a bottleneck in the computer system compared with processing capacity, thus, hindering the adoption of the LISP architecture.

On the other hand, a second line of research targets clean slate solutions as an alternative to the current Internet. As a result of these efforts, research projects such as GENI [26], 4WARD [27], DONA [28], TRILOGY [29], PSIRP [30] and TARIFA [31] have been developed to introduce novel network architectures. Overall, because these initiatives represent radical shifts of the networking paradigm new addressing schemes are further required. In the context of TARIFA—one of the most recent initiatives toward developing a clean slate Future Internet (FI)—the networking paradigm is no longer based on the interconnection of machines/interfaces but on the interconnection of services/resources—an approach that seems to meet much better both present and future user requirements. Accordingly, a new addressing scheme is required to make possible the shift to a service-oriented paradigm. This scheme must be able to overcome the semantic limitations already described for the current addressing architecture, and thus, need to be based on the locator/identifier separation paradigm.

## 1.2 Thesis Contributions

The work in this thesis contributes to enable semantic interoperability in two critical areas of networking, namely, network device configuration management and Internet addressing. To this end, our first contribution is to examine in detail why—despite numerous efforts—semantics remains an unsolved issue in the scope of both areas of networking. Taking into consideration the divergent nature of the semantic issues in each area, we approach these problems from different perspectives. Accordingly, specific thesis contributions derive for each area. These can be further divided into two major categories which are described next.

### 1.2.1 Semantic-Based Configuration Management

In the scope of network management, the work in this thesis contributes to the semantic issues in the configuration of network devices. We demonstrate for the first time the potential of Natural Language Processing (NLP) tools and semantic technologies for the abstraction and automation of CLI-based device configurations for heterogeneous

(i.e., multi-vendor) network infrastructures. The main contributions of this thesis in this regard are the following:

- We propose an Ontology-Based Information Extraction (OBIE) System which exploits natural language resources natively provided by network device vendors' in their CLIs and contribute to an architecture that can easily be generalized to other domains—wherein configuration of devices is also based on CLIs.
- We introduce a novel Information Extraction (IE) algorithm for CLI-based environments which enables automated and highly precise interpretation of CLIs configuration capabilities.
- We prove that the proposed IE algorithm yields the best system's performance by comparing against variations of our own algorithm and other authors' approaches and that our system performs in an efficient and highly accurate way.
- We formalize the semantics of the switch/router configuration domain using the Web Ontology Language (OWL) in an effort to become a conceptual reference model of the configuration knowledge in networking.
- We develop an online algorithm for seamless retrieval of configuration commands. The proposed algorithm enables outsourcing of configurations to third-party systems.
- We validate the proposed system in the context of a programmable and coordinated Network Management System (NMS) for multi-layer networks.
- We show that the large amount of information already available in the form of natural language text in CLIs is a valuable resource for automating and abstracting device configurations.

Overall—from a practical perspective—our solution provides network administrators with a simple tool which entirely automates and abstracts the complexities and heterogeneity of underlying configuration environments in order to reduce time and effort in the configuration of network devices. With such a tool, network administrators will no longer have to read hundreds of manuals, and configuration scripts can be automatically updated for new devices or system upgrades.

### 1.2.2 Semantic-Based Addressing

Regarding the semantic issues of the current Internet addressing architecture, the work in this thesis has a two-fold contribution. On the one hand, we contribute to a new

LISP Control Plane protocol capable of managing reachability and reliability issues in a LISP-compliant architecture, in an effort to improve technology feasibility. On the other hand, we propose a new addressing scheme for TARIFA—a clean slate service-oriented network architecture—based on the notion of semantic decoupling of host location and identification. Accordingly, the main contributions of this thesis are the following:

- We explore the major limitations of the current Internet addressing architecture and review recent literature in order to examine current strategies in the field. Furthermore, we identify open issues that hinder the deployment and adoption of these approaches.
- We propose a taxonomy for Identifier/Locator Split Architectures (ILSA).
- We propose the LISP Redundancy Protocol (LRP) as an approach for managing reachability and reliability issues of typical LISP architectures.
- We develop a novel addressing scheme for a service-oriented Future Internet architecture.
- We evaluate the performance of the proposed addressing scheme by developing an analytical methodology aimed to estimate the reachability and limitations of the identification space.

## 1.3 Supporting Publications

### Journals and Book Chapters

1. **A. Martínez**, M. Yannuzzi, J.E. López de Vergara, R. Serral-Gracià, W. Ramírez. "Applying Information Extraction for Abstracting and Automating CLI-Based Configuration of Network Devices in Heterogeneous Environments", to be published in *Artificial-Intelligence Applications in Information and Communication Technologies* (AIAICT 2015), Springer-Verlag in "Studies in Computational Intelligence Series" ISSN: 1860-949X.
2. **A. Martínez**, M. Yannuzzi, V. López, D. López, W. Ramírez, R. Serral-Gracià, X. Masip-Bruin, M. Maciejewski, J. Altmann, "Network Management Challenges and Trends in Multi-Layer and Multi-Vendor Settings for Carrier-Grade Networks", in *IEEE Communications Surveys and Tutorials*, 2014.
3. M. Yannuzzi, M. S. Siddiqui, A. Sallstrom, B. Pickering, R. Serral-Gracià, **A. Martínez**, W. Chen, S. Taylor, F. Benbadis, J. Leguay, E. Borrelli, I. Ormaetxea,

K. Campowsky, G. Giammatteo, G. Aristomenopoulos, S. Papavassiliou, T. Kuczynski, S. Zielinski, J. M. Seigneur, C. Ballester Lafuente, J. Johansson, X. Masip-Bruin, M. Caria, J. R. Ribeiro Junior, E. Salageanu, and J. Latanicki., "TEFIS: A Single Access Point for Conducting Multifaceted Experiments on Heterogeneous Test Facilities", in *Elsevier Computer Networks*, Volume 63, Pages 147–172, 22 April 2014.

4. W. Ramírez, X. Masip-Bruin, M. Yannuzzi, R. Serral-Gracià, **A. Martínez**, M. S. Siddiqui., "A survey and taxonomy of ID/Locator Split Architectures", in *Elsevier Computer Networks*, Volume 60, Pages 13-33, 26 February 2014.

5. R. Serral-Gracià, M. Yannuzzi, E. Marin-Tordera, **A. Martínez**, X. Masip-Bruin, "Metrics and QoE assessment in P2PTV applications", in *Int. J. Internet Protocol Technology*, Vol. 7, No. 3, Pages 148-164, 2013.

**Conferences and Workshops**

1. **A. Martínez**, M. Yannuzzi, J.E. López de Vergara, R. Serral-Gracià, W. Ramírez. "An Ontology-Based Information Extraction System for bridging the configuration gap in hybrid SDN environments", to be presented at *14th IFIP/IEEE International Symposium on Integrated Network Management* (IM 2015), IFIP/IEEE IM 2015 Conference Proceedings and IEEE Xplore®, Ottawa, Canada, May, 2015.

2. **A. Martínez**, M. Yannuzzi, R. Serral-Gracià and W. Ramírez, "Ontology-Based Information Extraction from the Configuration Command Line of Network Routers", presented at *Second International Conference on Mining Intelligence and Knowledge Exploration* (MIKE 2014), Lecture Notes in Artificial Intelligence (LNAI), Cork, Ireland, December, 2014.

3. W. Ramírez, X. Masip-Bruin, E. Marin-Tordera, M. Yannuzzi, **A. Martínez**, S. Sánchez-López, M.S. Siddiqui, and V. López, "A Techno-Economic Study of Network Coding Protection Schemes", presented at *Globecom* 2014 - Optical Networks and Systems Symposium (GC14 ONS), Austin, USA, December, 2014.

4. W. Ramírez, X. Masip-Bruin, E. Marin-Tordera, M. Yannuzzi, M.S. Siddiqui, **A. Martínez**, and V. López, "Improving Reliability in Multi-Layer Networks With Network Coding Protection", presented at *18th International Conference on Optical Network Design and Modeling* (ONDM 2014), Stockholm, Sweden, May 2014.

5. W. Ramírez, X. Masip-Bruin, M. Yannuzzi, D. Montero, **A. Martínez**, and V. López, "Network Coding-Based Protection Scheme for Elastic Optical Networks",

in *10th International Conference on Design of Reliable Communication Networks* (DRCN 2014), Ghent, Belgium, April, 2014.

6. W. Ramírez, X. Masip-Bruin, E. Marin-Tordera, M. Yannuzzi, **A. Martínez**, S. Sánchez-López, and V. López, "An Hybrid Prediction-based Routing Approach for Reducing Routing Inaccuracy in Optical Transport Networks", in *19th European Conference on Networks And Optical Communications* (NOC2014), Milan, Italy, June 2014.

7. R. Serral-Graciá, **A. Martínez**, E. Marin-Tordera, M. Yannuzzi, and X. Masip-Bruin, "Multi-Layer quality assessment framework for P2PTV applications," presented in *SaCoNeT ICC*, Ottawa, Canada, June 2012.

8. **A. Martínez**, X. Masip-Bruin, W. Ramírez, R. Serral-Graciá, E. Marin-Tordera, and M. Yannuzzi, "Toward a New Addressing Scheme for a Service-Centric Internet," presented in *SaCoNeT ICC*, Ottawa, Canada, June 2012.

9. **A. Martínez**, W. Ramírez, M. Germán, R. Serral-Graciá, E. Marín-Tordera, M. Yannuzzi, and X. Masip-Bruin, "An Approach to a Fault Tolerance LISP Architecture," in *9th International Conference on Wired/Wireless Internet Communications*, WWIC 2011, Vilanova i la Geltrú, Spain, June 15-17, 2011.

**Ongoing Publications**

1. **A. Martínez**, M. Yannuzzi, J.E. López de Vergara, R. Serral-Gracià, X. Masip-Bruin. "An Ontology-Based Information Extraction System for complementing network device configuration in the SDN future" (*IM 2015 Invited Extension*), to be submitted to IEEE Transactions on Network and Service Management (TNSM), April, 2015.

## 1.4 Thesis Structure

This thesis is structured in five main parts. Next, we provide a brief description of the content and topics discussed along each chapter.

**PART I**
**INTRODUCTION**

**Chapter 1** provides an introduction to the work developed in this thesis, including motivations, contributions, supporting publications and thesis structure.

## PART II
## BACKGROUND

**Chapter 2** examines in detail the interoperability challenges of managing multi-layer and multi-vendor carrier-grade networks, and review the current trends and recent standards in the area, with strong focus on industrial advances. We cover the Multi-Technology Operations System Interface (MTOSI) as well as OpenFlow, and analyze their potential impact and reach. We also discuss some of the reasons why relevant carrier-grade management proposals have not been able to fulfill the requirements of Internet Service Providers (ISPs), and identify a set of features that might help pave the way to market for new management products.

**Chapter 3** provides insights on the most important limitations of the current Internet addressing architecture, with main focus on the semantic overload of IP-based addresses. Moreover, we review existing proposals of ID/Locator Split Architectures (ILSAs) and delve into the Locator/Identifier Separation Protocol (LISP), in an effort to analyze its major limitations.

**Chapter 4** provides a review of the fundamentals in the field of Ontologies and Information Extraction (IE) as potential enablers of semantic interoperability for domains where standards have failed to find path.

## PART III
## SEMANTIC-BASED CONFIGURATION MANAGEMENT

**Chapter 5** describes the Ontology-Based Information Extraction (OBIE) System from the Command-Line Interface (CLI) of network devices, by introducing the rationale supporting our system and then, describing the system's architecture.

**Chapter 6** provides a comprehensive description of the Information Extraction algorithm, in an effort to understand how we enable automated interpretation of CLIs configuration capabilities.

**Chapter 7** describes the Command Retrieval algorithm which is responsible of performing the online functionality of our OBIE System.

**Chapter 8** presents the experimental framework along with the system's validation results. We also conclude on the various versions of our IE algorithm—when carried out over the configuration spaces of several widely used routers—and provide results on the performance of our system over enhanced CLIs.

## PART IV
## SEMANTIC-BASED ADDRESSING

**Chapter 9** describes the proposed LISP Redundancy Protocol (LRP)—a control plane protocol for the LISP architecture—aimed to provide reachability and reliability in scenarios of failure.

**Chapter 10** presents a novel addressing scheme for a service-oriented Future Internet Architecture. Moreover, we assess performance by estimating reachability and limitations of the identification space.

## PART V
## CONCLUSIONS AND FUTURE WORK

**Chapter 11** summarizes the main conclusions of this thesis.

**Chapter 12** provides the author's view on potential directions for extending the reach of the work developed in this thesis.

# Part II

# BACKGROUND

# Chapter 2

# Carrier-Grade Network Management

*This chapter aims to examine in detail the interoperability challenges of managing multi-layer and multi-vendor carrier grade networks, and review the current trends and recent standards in the area, with strong focus on industrial advances. We cover the Multi-Technology Operations System Interface (MTOSI) as well as OpenFlow, and analyze their potential impact and reach. We also discuss some of the reasons why relevant carrier-grade management proposals have not been able to fulfill the requirements of Internet Service Providers (ISPs), and identify a set of features that might help pave the way to market for new management products.*

## 2.1 Interoperability Issues in Multi-Layer and Multi-Vendor Carrier-Grade Networks

Network management has been the subject of study and one of the central targets of the telecommunications industry since the earliest days of networking. However, the overall management of multi-layer infrastructures still remains an open field for research, primarily because of the scarce interoperability between Network Management Systems (NMSs). In the context of multi-layer networks, management interoperability issues arise mainly due to the heterogeneity of technologies (i.e., IP and Optical) and the diversity of device manufacturers, coupled with the absence of standardized and broadly accepted mechanisms that enable both cross-layer and intra-layer communication of NMSs. A clear proof of this situation is the complexity of current Operations Support Systems (OSS). OSSs were originally conceived as the elements bridging business logic for

service provision and network management, but in practice they have had to incorporate several layers of "Umbrella NMSs" (as described later in section 2.2.1) to provide a minimum degree of workflow automation. Cutting down OSS complexity is among the most important efficiency objectives of practically all telecom operators.

The isolation between the IP and transport management ecosystems is exacerbated by the functional segmentation of standardization bodies, which are not necessarily working in the same direction. While, the Internet Engineering Task Force (IETF) is the organization playing a critical role in the scope of IP-driven network management standards [32], the International Telecommunication Union (ITU) [33], the Optical Internetworking Forum (OIF) [34], and the TeleManagement Forum (TMF) [35], are the most active organizations working toward the development of standards in the field of optical transport networks. In this section, we provide a detailed analysis on the Multi-Vendor Interoperability (MVI) and Multi-Layer Interoperability (MLI) issues for carrier-grade networks and outline the complexities that emerge as a result of the isolation of their management ecosystems.

### 2.1.1 Multi-Layer Interoperability (MLI) Issues

The MLI issues between the IP and transport NMSs arise primarily due to the development of systems targeting individual network layer functions in the Open Systems Interconnection (OSI) model. While the use of the OSI model has ensured that the technological diversity in lower layers does not affect the operation of the upper layers, the lack of mechanisms for enabling coordinated management between them has led to the replication of critical functions. Restoration mechanisms are a clear example of the redundancy produced by independent (i.e., per-layer) management functions in multi-layer settings, as they are featured by both layers to restore traffic onto alternate paths in case of failure. Indeed, the scarce efficiency for protecting and recovering multi-layer networks is the result of the absence of communication mechanisms between management ecosystems, which leads to the activation of restoration functions in individual layers

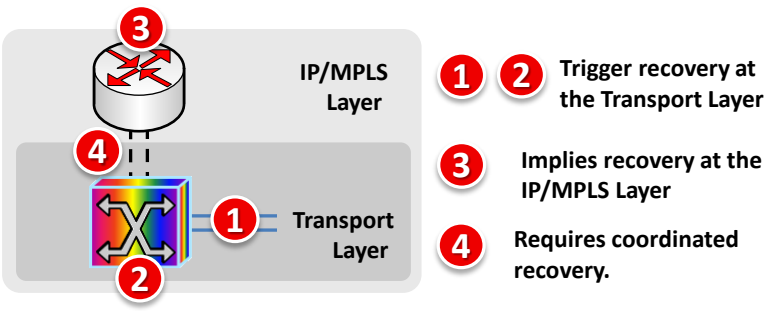


FIGURE 2.1: Typical failure scenarios in multi-layer networks.

based on the utilization of static time thresholds. In typical deployments, transport network restoration is attempted first, and restoration at the IP level is usually delayed by a static time threshold, which can lead to significant losses in case that restoration in the transport network fails [36].

Let us consider the example shown in Fig. 2.1, which summarizes the possible failure scenarios in a multi-layer network. Link and node failures at the transport layer (denoted as (*1*), and (*2*), respectively) can be typically recovered by the optical network without manual intervention, provided that the spare resources and capacity are sufficient for that end. A failure of a router (3), on the other hand, can be recovered by the IP/Multi-Protocol Label Switching (MPLS) layer. However, whenever a failure occurs at the interconnect point (4)—i.e., when an inter-layer failure occurs—a coordinated strategy is required for safe and efficient recovery. It is worth mentioning that, currently, this coordination is not dynamically resolved; quite on the contrary, it is predefined and meticulously pre-configured during network planning cycles. Observe that for failures (3) and (4), if no backup router is available at the IP layer, the transport layer could reactively attempt to optically bypass the affected router. Unfortunately, the lack of cross-layer management mechanisms does not allow dynamic cross-layer recovery, which leads to an unnecessary duplication of resources and restoration mechanisms at both layers. Indeed, enabling communication between NMSs of different layers would derive in much more efficient restoration strategies, and could help alleviating in the future part of the duplicity in current carrier-grade protection schemes.

Another important aspect is that the MLI issue also hinders the automation of multi-layer tasks between the management ecosystems of the different network layers. Hence, only manual means are attainable for coordinating cross-layer operations. By manual means we refer to the human interactions between administrators in each network domain. To illustrate this, consider the multi-layer setting shown in Fig. 2.2. In practice, for setting up an IP service (e.g., the provision of IP links), the Planning or Sales Department must first issue a new service request to the IP Network Management Department (1), which in turn performs the corresponding actions to check for the availability of IP resources (2). If the required resources are available at the IP layer, then the administrator will issue a request to the Transport Network Management Department for the set up of new optical paths (3). It is worth highlighting that, as the IP and Transport networks are typically managed separately and independently, they require to comply with formalities and procedures for information exchange, a fact that often results in high provisioning timescales. Once the request is properly received by the Transport Network Management Department, meaning by this that the request satisfies all the requirements from both parts, the Transport Department completes a series of actions, including confirmation of available resources and their subsequent provisioning at the

transport network (4). After the Transport Department has fulfilled the incoming request, an acknowledgement is sent back to the IP Department (5). From this response, the IP Network Management Department completes the corresponding configurations of IP devices in their domains (6), and finally, notifies the completeness of the IP service request (7). As seen in Fig. 2.2, the manual coordination process is very slow, so even the simplest cross-layer operation, such as the establishment of a single IP link, can take hours or even days. As depicted in Fig. 2.3, in large carrier-grade networks routers $A$ and $B$ may perfectly belong to different "IP management domains" (e.g., when router $A$ is supplied by router vendor $V_A$ and router $B$ by vendor $V_B$). Likewise, Fig. 2.3 also shows that, the lightpath required in the transport layer for supporting the IP link may also need to traverse different "Transport management domains" (e.g., from vendor $V_X \rightarrow V_Y \rightarrow V_Z$).

In the scope of cross-layer operations for multi-layer networks, services that could be provisioned in a scale of minutes, currently range to the scale of days or weeks, a fact that beyond its technical and functional implications translates into higher operational expenditures (OPEX). Moreover, the configuration of fixed or dynamic policies for proactive cross-layer operations (e.g., for dynamic traffic management) are not openly supported

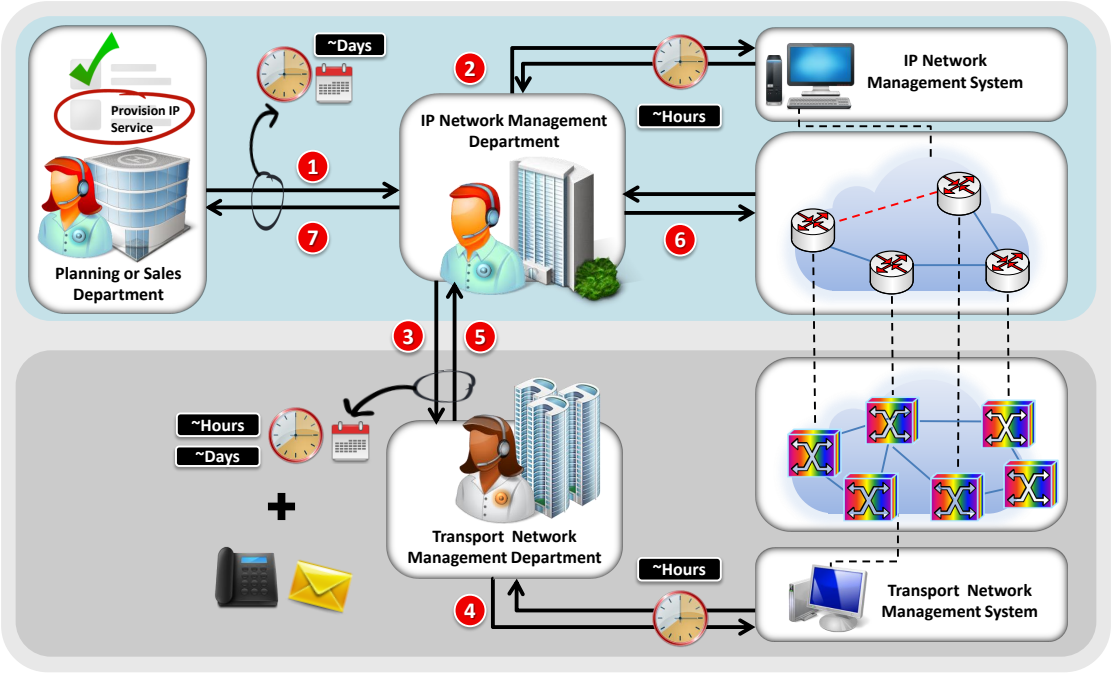


FIGURE 2.2: Current workflow for provisioning an IP service involving configurations at both layers.

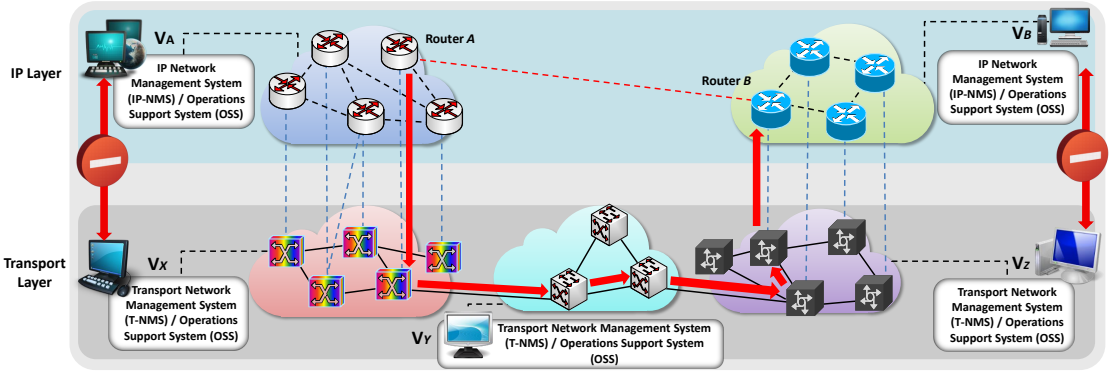


FIGURE 2.3: An example of network management interoperability issues involving cross-layer resource provisioning of an IP link between routers $A$ and $B$ of a carrier network.

by current multi-layer settings[1]. Indeed, if more advanced cross-layer operations were developed, they could further contribute to make significant progress in aspects such as multi-layer recovery and self-healing. By self-healing we mean coordinated protection mechanisms that could provide network reliability and automated restoration actions to recover from unplanned failures. Moreover, another significant challenge is the automatic discovery of interlayer connections, as for now network administrators rely on manually built topological databases. Finally, the advent of third party systems makes integration to multi-layer environments more important than ever, as current external tools could be combined to automatically interact with the multi-layer environment, so the latter can benefit from the services of external subsystems, such as multi-layer Path Computation Elements (PCE) [39], monitoring tools, accounting systems, etc.

Accordingly, the absence of standard management interfaces for inter-layer communication not only results in duplication of network functions and lack of automation for cross-layer provisioning tasks, but also derives in slow provisioning timescales, lack of proactive policy-based and failure-related interactions, as well as limited integration of third party management subsystems [40].

Although several efforts are underway for overcoming the MLI issues described above, the overall progress is slow. Most of the solutions proposed thus far have not had enough echo among ISPs, so it is hard to assert that there is a clear trend toward tackling the MLI issues at a management level. In Sections III and IV, we will provide an analysis of the main limitations of current research trends, in order to set the ground for future research lines in the area.

[1]There are some proprietary solutions, but they are naturally constrained to single vendor settings, for example, Juniper's solution based on hybrid nodes [37] or Huawei's Hybrid MSTP OSN 7500 II [38]. In Section IV, we will cover some of these solutions.

### 2.1.2 Multi-Vendor Interoperability (MVI) Issues

Various bodies such as the International Telecommunication Union (ITU-T) and the Internet Engineering Task Force (IETF) are working toward developing standard specifications of control and data plane functions enabling interoperability between different vendors. However, in practice vendors compete with each other trying to gain higher market share by implementing specific non-standard functionalities in their equipment, a fact that unquestionably leads to customer lock-in. This in turn creates a buyers dependency on the seller. According to the report released by the market research firm *Infonetics Research* in the second quarter of 2013 [41], Cisco had the highest market share for carrier router and switches, followed by Huawei who has gained the most market share over the past two years. Overall, Alcatel-Lucent, Cisco, Huawei and Juniper account for 90% of the market share, while the remaining 10% is split among several other equipment vendors.

The demand for new functionalities has led to vendor-interoperability requirements in the data as well as the control and management planes. A classical example of interoperability challenges in the data-plane can be seen in the 100G optical solutions available in the market today. While standards propose the use of different configurations of single 100G or multiplexed 40G and 10G lightpaths using fixed spectrum slots of 50 GHz, vendors such as Ciena, have developed custom 100G optical solutions that employ larger spectrum slots to provide longer reach, but makes it limited in terms of interoperability with other vendors' equipment.

MVI issues related to the management plane are primarily reflected as the lack of standardized interfaces for communication between NMSs from different vendors within a single layer. Let us consider the following example to illustrate the MVI problem. As shown in Fig. 2.3, nowadays, when a service provider requires to configure a service spanning across different vendor (management) domains (e.g., between Ciena $[V_X]$ and Huawei $[V_Y]$ at the transport layer) the configuration process is done ad-hoc, mainly due to the lack of means for interoperation between NMSs, which translates into high operational and maintenance expenditures. The forces driving service providers to buy both Ciena and Huawei are two-fold: firstly, better costs when negotiating prices (lower capital expenditures - CAPEX) and secondly, avoid single vendor dependency (which lowers CAPEX but increases OPEX). This is a typical example of the MVI issue, a problem suffered by almost every service provider, and which will remain present while purchase policies remain the same. Another scenario that reveals such incompatibility between NMSs is the designation of skilled software development teams, committed to develop dedicated software agents to enable interoperability between NMSs and network devices

of different vendors. For instance, this case is well-known at ADVA Optical Networking, where the management interoperability issues with Nokia-compliant technology are tackled through ad hoc software developments.

Herein, we refer to the MVI problem in two dimensions, firstly the communication between NMSs in a single network layer (i.e., within IP or within the Transport layer) and secondly, the communication between a NMS and multi-vendor devices (i.e., routers, Optical Cross Connects (OXCs) or any other network elements), for example, for performing configuration tasks. This dimension of the MVI problem has led to the development of standard open interfaces, thereby enabling hardware independency (e.g., OpenFlow)—a hot topic in the field which will be further discussed in Section 2.5.

As for the first dimension of this problem, despite the numerous efforts for providing *multi-vendor support*, at present many network management software solutions are constrained to interoperate with other management systems, a problem which normally leads to high stresses in internal implementations and increased costs [42]. As mentioned previously, proprietary network management protocols as well as vendor-specific management data representations are the hallmark of many manufacturers to distinguish from other vendors. However, these business and market driven decisions result in serious interoperability limitations for managing heterogeneous networks. This problem goes beyond the scope of standard protocols for network management and steps into the field of device-vendors, many of which strongly reject the belief of being as good at managing the competition's products as their own [43].

As for the second dimension of the MVI problem, let us consider the following example. In IP networks, the primary interface used for configuration is the Command Line Interface (CLI), a vendor-specific technology which can even change between devices of a single vendor, e.g., according to the Operating System (OS) version used. Due to the customization degree per vendor, under this scenario, achieving truly integration of multi-vendor device configurations is a rather challenging task. Even in the presence of standardized communication protocols and mechanisms, the data models used by different vendors can vary significantly giving place to interoperability challenges. For example, the Simple Network Management Protocol (SNMP) [44] [45] is used as a standard protocol to communicate with devices in order to facilitate inventory management, alarm notification, and performance monitoring. However, different vendors specify extended Management Information Bases (MIBs) for SNMP to facilitate advanced features that are not described in the standard MIBs. This means that operators have to modify their management systems whenever they introduce new devices in the network or change devices to a different vendor. The explanation for the prevalence of proprietary approaches in industry, is that they are the result of forces driven

by vendors rather than by service providers, thus, it highlights the divergent interest of both parties. While operators target the interoperability in heterogeneous networks, manufacturers clearly target their own business interests.

Two technologies, the Network Configuration Protocol (NETCONF) [10] and the Multi-Technology Operations System Interface (MTOSI) [46] are positioned as promising technologies for the IP and Transport Network Management ecosystems, respectively. On the one hand, NETCONF seems the expected standard for remote IP device configuration. This protocol aims to overcome the limitations of SNMP-based configurations as well as proprietary CLIs to reduce complexity and lower OPEX—according to industry sources network device configuration is actually the greatest contributor to OPEX [12]. NETCONF is defined for providing configuration state maintenance, concurrent configuration, transaction across devices and roll-back support. Initial implementations of NETCONF in vendor equipment have already been developed (e.g., see Juniper, Cisco and tail-f [47, 48, 49]). Nevertheless, it is important to note that NETCONF only foresees the access protocol and configuration of network elements, but it does not take into account the definition of configuration information, i.e., an explicit way of expressing its payload.

For a complete definition of the configuration model, a data modeling language is also required, to provide the means for defining and declaring the network elements particularities (e.g., interfaces, addressing, bandwidth, etc.). In other words, a concrete and precise way of expressing what can be read and written over the configuration is needed. The lack of a standard data model across all vendors led to a new challenge around NETCONF. Indeed, even if NETCONF is implemented by all vendors' equipment, its future depends on the adoption of a standard modeling language, allowing a common view and knowledge of network equipment. The initial implementations of NETCONF relied on proprietary data models, which in turn raised clear interoperability issues. Examples of data models that have been proposed and tested for its use within the NETCONF framework are: the XML Schema (XSD) [50], Relax NG [51] and Ontology Web Language (OWL) [52].

More specifically, Extensible Markup Language-based (XML) languages, such as XSD and Relax NG, were initially considered as potential candidates for NETCONF content definition. They were evaluated and compared with YANG [13], in order to assess their suitability as network management data models [51, 53]. These studies compared language data models according to their level of expressiveness, elements of construction, readability and interoperability, and revealed that XML Schema languages were suitable for general constructions, but they had neither the adaptability to model hierarchical data in a clear and concise way, nor the level of expressiveness provided by YANG [13]—a

data model that has been specifically designed for the NETCONF protocol. According to these studies, YANG also outperformed schema languages in terms of readability, due to its likeliness to programming languages. Furthermore, XSD and Relax NG were shown to be too general for domain-specific data modeling in the context of network management.

On the other hand, OWL [54]—a W3C specification for authoring ontologies—was also proposed as a modeling language for network management information [52]. The proposal is to define OWL modules for common concepts of any NETCONF configuration model, as well as operations and notifications, and to agree on a standard serialization method, while suggesting RDF/XML for this purpose. The initiative proposed in [52] also exposes how OWL fulfills NETCONF's main requirements, such as the capability to define NETCONF operations and newly derived ones, error annotation capabilities, human readability, meta-data support, reusability, support to basic types and relationships, etc. Nevertheless, OWL as well as other XML-based initial approaches have failed to position as the best candidates for network configuration data definition either because of their lack of semantics or their intricate use.

At present, YANG [13] is positioned as the strongest candidate to a standard data model language for NETCONF. It is the IETF's proposed standard to create a common language for data modeling definition, and although some YANG compliant software applications have already been developed, there is still a long path before YANG and NETCONF become leading standards in the network configuration arena.

As for the transport ecosystem, MTOSI emerges as a protocol aimed to overcome the interoperability issues of SNMP and it operates with unified network data models and operations supported by Web Services. However, a limitation is envisioned for MTOSI. Despite the fact that there is a standardized data model specification, the representation of devices within this data model is different, thus, vendor interoperability becomes a major challenge. In Section 2.5, we will delve into the potential of MTOSI for enabling interoperability in multi-layer infrastructures.

#### 2.1.2.1 The Semantic Problem in Network Device Configuration Management

Indeed, the task of network configuration has become one of the most critical and complex areas in network management [12]. Network configuration typically deals with the maintenance, setup, repair and expansion of services and network functions. It is performed for multiple reasons, but overall, it aims to deploy end-to-end network services,

ensure performance, minimize downtime, support rollback in case of failures, enable device software management as well as collect configuration data. Notice that, in general, the term *network configuration* refers to the configuration of the network as a whole, for example, the deployment of a Virtual Private Network (VPN) service is a typical configuration task, usually expressed through high-level requirements which translate into low-level (individual) device configurations. Herein, our focus is on the configuration of network devices rather than the network as a whole.

As previously stated, the lack of standard protocols for network device configuration makes network management increasingly complex for network administrators—particularly in the context of heterogeneous network infrastructures. In this context, the absence of standard protocols has prompted the use of proprietary mechanisms for the configuration of network devices. The Command-Line Interface (CLI) is in fact the preferred mechanism for network device configuration—near 95% of current network devices are configured through proprietary CLIs [12]. Despite its widespread use for configuration purposes, CLIs have numerous limitations, some of which we will briefly overview.

***Proprietary Protocol.*** The most significant limitation of CLIs is their *proprietary* nature, i.e., CLI-based configuration environments are not standard, and thus, are specific to each vendor. This is exacerbated by the fact that within each vendor's space a CLI can also be specific to a device model or operating system version. This means that, the terminology, commands, configuration operations and related concepts can be dissimilar even among devices of a single vendor. In order to cope with the ever changing configuration environment, network administrators must develop advanced skills, gain specialized knowledge and continuously update to encompass the full range of devices available in the network market. Figure 2.4 depicts an extract of the CLI environment

```
Telnet 10.1.2.15

configure
set interfaces <interface-name> unit <interface-unit-number>
family inet address <source>
commit
exit
```

(a) Router Vendor: Juniper.

```
Telnet 10.1.1.20

enable
configure terminal
interface <interface_id> <slot number/port number>
ip address <ip_address> <mask>
no shutdown
end
```

(b) Router Vendor: Cisco.

FIGURE 2.4: CLI commands for the configuration of an access control list on two different router vendors.

for two different vendors (Juniper and Cisco, respectively). From this figure we can observe that different sets of terminologies and commands are used by each vendor to refer to the *same* configuration operation, namely, configuration of an IP standard Access Control List (ACL). Notice that, not only commands differ syntactically but the granularity and arrangement of the hierarchy also differs among vendors.

***Lack of semantic interoperability.*** Furthermore, the foundations of the configuration problems are not restricted to syntactical differences, but most importantly, they extend to semantic dissimilarities between CLIs as well. Due to their proprietary nature, device vendors customize their CLIs as a way to unequivocally distinguish from their competitors (as shown in Fig. 2.4). This leads to definitions of the configuration space in their own terms. In some cases vendors set the hallmark by launching their own terminology, while in other cases their interpretation of the domain leads to misleading use of terms with respect to the common routing domain knowledge, or even overlapping meanings in relation to different terminologies with other vendor configuration spaces. The lack of common standards for the conceptualization of the routing domain has led to scenarios wherein the same terms are used to refer to different concepts or where different sets of terminologies have the same meaning. As CLIs were devised for human operators, in practice, this issue is addressed by featuring help descriptors which aim to resolve the semantics of the configuration domain by providing users with a concise natural language description of commands and variables. Generally, help descriptors are a way to narrow down to the common and shared conceptualizations of the domain of knowledge and guide users through the configuration process. In this context, help descriptors are unquestionably valuable resources for network administrators as a means to disambiguate the semantics of configuration commands. However, in the context of automated environments, there are no means for achieving interoperability at a semantic level.

***Human/User-driven.*** The CLI was ideally designed for user-driven operation, i.e., manual-based configuration. This explains its text-based interface and natural language help feature. Moreover, the heterogeneity of CLIs also hinders the automation of network configuration in multi-vendor networks. The only means to achieve automation to a certain extent relies on the use of configuration scripts, an approach that aims to simplify recurrent configuration tasks. However, scripting is neither scalable nor practical as network administrators must adjust and fine-tune their configurations for each system upgrade or network equipment purchase. For this reason, CLI-based configuration is rather challenging and mostly performed in a manual way. Manual-based configurations are likely undesired, not only because they are error-prone and increasingly complex,

but mostly because they significantly increase OPEX.

***High Operational Expenditures.*** Performing network device configuration through CLI-based mechanisms entails high OPEX. Network device configuration is considered in fact the major contributor to OPEX for current telecom providers [12]. This stems from the fact that manual configuration is expensive. The delay in service provisioning naturally leads to increased costs, in addition to the costs that carry network outages whenever there are mistakes in the configuration process and the expenses required to continuously train network administrators in the configuration of each vendors' space.

***Error-prone.*** Configuration errors can lead to network outages, performance degradation and increased OPEX [55]. CLI-based configurations are highly sensitive to errors due to their manual nature, and note that this can potentially affect other network devices or the network as a whole.

According to the previous facts, the proprietary, user-driven and manual-based nature of CLI-based mechanisms make this technology an unsuitable solution to automate network device configuration in the context of heterogeneous networks. Overall, the issue with CLI-based mechanisms is the lack of semantic interoperability between configuration spaces. The means for semantically defining the configuration domain are solely expressed in the form of natural language textual resources targeting human network operators.

## 2.2 Traditional Strategies to face the MLI and MVI Problem in Network Management

Initial efforts led by industry and academia to overcome the MLI and MVI problems in network management have put into practice several strategies. In-house developments, IP over Dense Wavelength-Division Multiplexing (IPoDWDM) [56], and control plane technologies [57, 58], are some of the main solutions currently deployed by many large ISPs with the aim of simplifying the missing management functions in multi-layer and multi-vendor settings. Nevertheless, these developments either solve partial problems or target them in a temporal and local way, hindering the possibilities of becoming absolute trends in the field of multi-layer network management. Either way, they provide valuable foundations for: (i) providing basic support of missing management functions, (ii) envisioning new requirements for future approaches targeting interoperability and

(iii) bringing to light the real needs of network operators as they evidence the existence of numerous interoperability problems.

### 2.2.1 In-House Developments

As stated in the previous section, the choice of large telecommunication service providers to defeat mono-vendor settings is usually a business and market-driven decision, which has naturally led to heterogeneous network infrastructures. The problems arising from such inherent heterogeneity have resulted in the development of in-house applications, aimed to locally address the management limitations in multi-vendor and multi-layer settings (see Fig. 2.5). These platforms are developed as internal network management engines and are also known as Umbrella NMSs. They are custom-made and they are typically centralized solutions built to temporally address the specific needs of network administrators. They follow a Manager of Manager (MoM) architecture, an alternative that has always been available for network managers to "smooth" the limitations around existing management solutions.

Umbrella NMSs lack of flexibility, efficiency or means for exchanging or enabling interoperability between layers in a multi-layer setting. They usually perform as central systems with no top-to-low layer communication or vice-versa. As shown in Fig. 2.5, these umbrella systems typically offer custom interfaces to existing NMSs developed in each network layer *(A)*. In many other cases, the umbrella NMSs directly interact with the network elements *(B)* to avoid the complexity, the high costs and administrative delays that derive from requesting new functionalities to be developed over proprietary NMSs. This allows to achieve the required management functions at lower costs, reduced time-scales, and higher levels of customization. In-house developments are not efficient and have brought to light the inadequacies of current network management tools for targeting the problems that arise in the context of multi-layer settings. Umbrella NMSs are just an example that discloses the fact that current NMSs lag far behind the requirements faced in practice by network managers.

### 2.2.2 IP over DWDM

IPoDWDM [59, 60] is a solution for approaching the core network architecture in order to support the ever-increasing volume of IP traffic. This architecture proposes the convergence of IP and DWDM transport technology by integrating coloured transport interfaces directly in the IP routers and photonic switching into DWDM platforms. Integration of transponders onto routing platforms eliminates external layers of transponders between the IP and optical transport layers. The overall goal is to simplify the

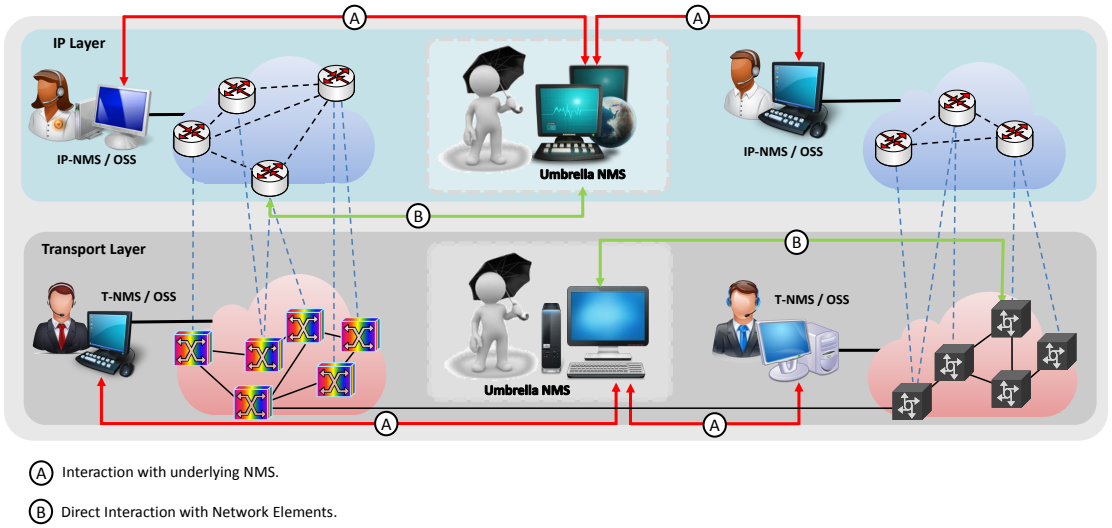


FIGURE 2.5: Facing the interoperability issues with in-house developments.

network while lowering costs. This integration highly contributes to capital and operational savings due to the reduction in the number of network elements, space occupation, and energy consumption. It also leads to simplification of the network by eliminating Synchronous Digital Hierarchy/Synchronous Optical Networking (SDH/SONET) boxes. Furthermore, it provides inherent protection capabilities, due to the knowledge shared between layers, as IP devices can monitor the optical path. However, the final goal for full multi-layer integration assumes support of the control plane to complete cross-layer network provisioning and monitoring—we will delve into these aspects in the following section.

### 2.2.3 Control Plane Technologies

Several network device suppliers already provide Generalized Multi-Protocol Label Switching (GMPLS) support in their carrier-grade equipment (e.g., Huawei's OptiX OSN 9800, Juniper's M-series or Cisco's CRS-1/3), and some carriers have even fostered the early adoption of this technology. For instance, the major operator in Japan, namely, NTT, has been a leading promoter of the GMPLS technology [61]. Despite these advances, technologies such as GMPLS are still not widely deployed in practice [62]. The true fact is that telecom providers are not under a big pressure for deploying GMPLS—at least not until the latter acquires more momentum. This means that, in the meanwhile, IPoDWDM solutions are restricted to scenarios where mainly manual transport paths are established, arising clear interoperability issues between layers. Indeed, the proprietary nature of most IPoDWDM solutions, conditions the network infrastructure in many cases to mono-vendor settings. Integrated management under this particular scenario would be restricted to end-to-end provisioning for a single-vendor solution.

In addition, signal compatibility is also a limitation of these solutions, since standardization efforts are still required to ensure a perfect match between transponders and transport equipment. Accordingly, the value of these solutions is limited in scope to signal compatibility and interoperability constraints. The lack of standardization at the signal transmitted by the transponder is an important issue in the integration of transponders at the IP/MPLS cards [56]. There are some standardization efforts at the ITU G698.2 [63] to solve the problems of the so called "Black Link". A Black Link is defined as a link where the transponders (source and destination) and the intermediate optical equipment may be from the same or from different vendors. This standard defines the signal parameters for the input and output interfaces in such network. There are two different kinds of wavelength connections: (1) "Friendly Wavelength" and (2) "Alien Wavelength". A Friendly Wavelength is a lambda connection not created in the optical system, but it is known and managed by the optical management system. This Friendly Wavelength may be created by a router, but the optical management system exchanges information with the router to know the status of the connection. Usually this is done in mono-vendor solutions with virtual transponders from the optical management system point of view. On the other hand, an Alien Wavelength is a connection created outside the optical domain (e.g., from an IP Router), but it is not managed by the optical NMS, hence, the DWDM system has no early knowledge of signal parameters (e.g., bandwidth, wavelength). In such scenario, the optical management system is signaled to configure the intermediate OXCs, but it does not have information about the signal quality. Hence, intermediate integrated solutions open new issues such as the optical signal information exchange between the IP/MPLS and the optical management systems.

Overall, IPoDWDM strategies have been driven by several device manufacturers (e.g., Cisco, Juniper and Alcatel-Lucent) setting the trend of next generation networks. Whether the management of a network with IP/WDM integration with coloured interfaces is simpler than todays network management is still an open issue, as most routers do not deal with transport specific issues and some problems are still open to resolution.

## 2.3 New Trends in Multi-Layer Network Management and Challenges Faced

Aside from the strategies traditionally deployed by service providers to face the MLI and MVI problems, recently, new trends have emerged in the field of multi-layer network management. These solutions follow one of two lines. They either represent new ways of approaching the multi-layer network management problem [64, 65] or they feature

new emerging technologies [37, 66], which demand new forms of interaction and pose different challenges from the management point of view.

While service providers pursue new trends for targeting the convergence of IP and Optical layers, research efforts shall consider all the newly emerging challenges and complexities in order to enable interaction between layers at the highest level of abstraction. Thus, solutions within the scope of multi-layer network management should provide the flexibility for delivering true full systems not compromised to specific technologies.

To this end, we present an analysis on new trends in the scope of multi-layer networks. We firstly introduce MPLS-TP a profile of MPLS for transport networks, then we comment on various solutions based on integrated NMSs for multi-layer networks, such as the solution developed by Cyan [64], and finally, we delve into hybrid solutions for achieving IP over Optical integration, e.g., Juniper's PTX [37] or Huawei's Hybrid MSTP OSN 7500 II [38].

### 2.3.1 MPLS-Transport Profile (MPLS-TP)

Multi-Protocol Label Switching Transport Profile (MPLS-TP) [65] has been built as a joint effort between ITU-T and IETF. Its goal is to extend and enhance the concepts of MPLS [67] and create a profile that can be used in the context of transport networks. MPLS-TP not only adds a set of extensions to MPLS, but also disables those capabilities neither required nor consistent with transport technologies. An example of such suppressed capabilities are Label Switched Path (LSP) merge [68], Penultimate Hop Popping (PHP) [69] and Equal Cost Multi Path (ECMP) [70]. These features are removed because they hinder the Operation, Administration and Maintenance (OAM) functions, which are mandatory for transport networks. For example, the PHP functionality, which consists on removing the label at the penultimate hop—i.e., before reaching its destination—would remove significant information required by OAM functions to work accurately. Another major difference between both technologies is that, unlike MPLS, MPLS-TP LSPs are bidirectional, therefore, the forward and backward LSPs follow the same path. This provides the ability to communicate back to the source if any problem is encountered, thus simplifying troubleshooting.

In a nutshell, MPLS-TP aims to enable MPLS on transport networks, while keeping their operation as close as possible to already existing transport technologies (e.g., SDH/SONET networks). To this end, MPLS-TP—like is the case of most transport network platforms—is not constraint to conveying IP packets, meaning that, MPLS-TP functionality can be fully provided regardless of the packet-layer technology used. MPLS-TP has been designed to provide a rich set of control plane-independent OAM features. In

MPLS-TP all OAM functionalities run inband, hence, OAM packets are sent over the same path of the user payload in the MPLS-TP forwarding plane to manage and monitor the transport network and the services being delivered. Some of the OAM features include, Continuity Check (CC), Connectivity Verification (CV), delay and loss measurements as well as fault notification. An overview of the MPLS-TP OAM framework can be found in [71].

In addition to OAM enhancements, MPLS-TP provides two modes of operation. Firstly, configuration of LSPs based on network management plane technologies, which are usually referred to as "static" configuration (i.e., not based on a dynamic control-plane), and secondly, dynamic provisioning based on control plane technologies (e.g., GMPLS). The main advantage of the former approach is that MPLS-TP networks can be managed through a centralized NMS as in traditional transport environments (e.g., in Reconfigurable Optical Add Drop Multiplexers - ROADM-based Optical Networks), which makes it quite appealing for network operators who are used to manage and provision their services in that way. The latter approach, suggests the use of the GMPLS control plane to provide automatic setup of MPLS-TP LSPs. For more details on MPLS-TP's components and its technical specifications the reader is referred to [65, 71, 72, 73, 74].

In light of all this, MPLS-TP is foreseen as a technology to leverage traditional transport technology to support packet-based services in a more efficient way, while embracing the features of traditional transport environments (e.g., QoS, centralized operation, etc.). Over the recent years, network device manufacturers and network operators have progressively begun to provide support to standard-based MPLS-TP solutions (e.g., Cisco, Nokia Siemens Networks, ZTE, Verizon, Bharti Airtel, Huawei, Telecom Italia, China Mobile, etc. [75]), while early applications have already been discussed and deployed. Service Providers are considering the migration of traditional SDH/SONET networks to packet transport technology. Along this path, MPLS-TP appears to be a well-suited technology to overcome the inefficiency of these networks when transporting packets—basically due to constant bit rates even in the absence of traffic. Indeed, plans for deploying dynamic MPLS-TP over OTN/DWDM in Verizon's core network were reported in [75]. The goal was to provide connectivity to edge services by interconnecting Ethernet and IP/MPLS domains through MPLS-TP.

Furthermore, MPLS-TP has been considered a candidate for replacing Ethernet in the access network. Data plane compatibility between MPLS-TP and IP/MPLS could make interoperability with the backbone of large ISPs rather simple, while providing end-to-end OAM. In this context, the selling point is that by deploying MPLS-TP—along with already existing deployments of IP/MPLS—ISPs will have simple and consistent ways of provisioning and managing their networks edge-to-edge. However, technical comparisons

in 2012 between both technologies seem to favor Ethernet, claiming that MPLS-TP still lacks of maturity and security to yet dominate access networks [76]. If MPLS-TP is ready to replace Ethernet or other packet transport technologies in the access or the core network is yet to be seen.

Overall, one of the driving forces of MPLS-TP is the possibility of deploying a unified MPLS strategy, wherein ISPs can use MPLS in its different flavors (i.e., IP/MPLS, MPLS-TP, etc.) from core to aggregation and access to improve end-to-end convergence. At the same time, the dual mode of operation of MPLS-TP brings flexibility into the design of transport networks as static configurations may be used while control plane technologies are not yet in place or required. However, the requirements in the scope of network management go beyond the provisioning functionality and instead require even more complex interactions (e.g., scheduling, alarm correlation, and self-healing, among others) to really address the key issues in multi-layer network management.

### 2.3.2 Integrated Multi-Layer Network Management Systems

An integrated NMS for multi-layer networks has been for some time among the research interests of academics [77, 78, 79] and most recently launched by Cyan under the figure of Cyan's Multi-Layer Management System (CyMS) [64]. The aim of these solutions is to provide a unified network management plane to effectively manage (i.e., configure, provision, monitor, etc.) multi-layer networks regardless of the technological differences.

Early works such as [78] and [79] proposed an integrated NMS capable of providing multi-layer connectivity services based on the use of management functions supported by single layer control plane technologies—whenever available. This approach mainly focused on supporting basic configuration management functions to provide end-to-end IP connectivity derived from Service Level Agreements (SLAs). In addition, some of the same authors devised a policy-based architecture [80] supporting security policy definition, storage and enforcement. Moreover, the authors in [77] developed a prototype implementation of an integrated platform based on standard technologies, to address the issues of multi-layer network management. In that research initiative, the authors defined their own XML-based management information modeling language, adopted a new XML-based management protocol and developed mediation modules to interact with legacy protocols (e.g., SNMP). They also featured policy-based management capabilities while providing an open interface through which basic support to services was provided. That prototype was assessed under a simulation environment to prove support to multi-layer network optimization tasks, fault management and restoration functionality.

Beyond the scope of research, Cyan has actually commercialized its product release of an integrated network management solution, namely, CyMS. CyMS provides a software solution for integrated three-dimensional view of the physical and logical connections in all network layers. It is adapted and designed to comply with TMF Multi-Technology Network Management (MTNM) and ITU G.800 principles [81]. Additionally, it features three-dimensional heat maps for enabling proactive network monitoring and control, while gradient-color coding indicates the existence of critical conditions—for instance, for anticipating future faults or service degradation. Indeed, CyMS provides enhanced functionalities for Cyan-compliant devices. However, its potential and dynamics is limited to its own optical gear. This proprietary solution is restricted in scope to multi-vendor environments, a fact that makes it an unsuitable solution in the context of most common and heterogeneous network deployments. The constrained multi-vendor reach also contributes to high costs of maintenance and updates due to new technologies or requirements.

### 2.3.3 Hybrid Node

Several hybrid solutions drawn from industry and academia have emerged in the field of multi-layer networks [37, 38, 82, 83]. The Juniper PTX converged supercore [37] unveiled on March 2011 is Juniper's hybrid solution to multi-layer networks. Juniper's PTX stands for "packet transport switch", and, unlike the IPoDWDM solutions described earlier in Section 2.2, this hybrid approach takes integration a step further by combining optical and electronic technologies in a unique box. The combination of hardware and software to develop a single hybrid solution aims to provide integration of IP/MPLS and optical control and management planes. This integration allows to enable coordinated provisioning, planning and modeling, avoiding traditional duplication of management functions. This transport strategy assumes that IP and Optical layers will no longer be isolated ecosystems. In this sense, Juniper's PTX points in the direction of an integrated approach for multi-layer network management. The convergence of packet/optical layers within a single platform derives in seamless integration of their control planes based on GMPLS technology and User-Network Interface (UNI+), which facilitates multi-layer provisioning, management and restoration. However, an integrated approach aligned to the foundations of Juniper's PTX converged supercore has a number of limitations: (i) integration is constrained to a single vendor—at least with current available technology, hence not solving the MVI problem; (ii) it represents a disruptive approach which requires of new network infrastructures; and (iii) ISPs will

quite likely keep buying optical equipment to other transport vendors as well as routing devices to other IP manufacturers, meaning that, this would also have important implications from the business and operational points of view.

Moreover, other hybrid approaches have been researched in academia [82, 83]. The authors in [83] developed a hybrid optoelectronic router in which optical and electrical technologies and complementary metal-oxide-semiconductor (CMOS) electronics are combined into a single router. They demonstrated that such an architecture enables reduction of power consumption and latency while still providing the capabilities required in optical packet switching. R. Cafini et. al. [82] proposed a modular programmable router architecture to provide dynamic management and configuration of services and resources. Indeed, a relevant contribution of this work is to consider *network programmability* to achieve dynamic network layer functions. The value of featuring network programmability is that it has actually become a must for next-generation networks, and is the main driver for Software Defined Networking (SDN)—more details on network programmability and SDNs as a means to provide flexibility and openness for network infrastructures are given in Section 2.5.3. Certainly all these approaches for developing hybrid devices are aligned with the belief of improving network management and resource consumption while integrating the management and control planes of different network layers. Conversely, coordinated strategies diverge from this view, based on the belief that an intermediate system should "coordinate" functions in multi-layer environments. In Section 2.5, we will delve into the future trends in multi-layer network management, and contrast integrated approaches against coordinated ones.

## 2.4 The Role of Third Party Management Subsystems in Multi-Layer Networks

The advent of complementary management subsystems has enabled specific cross-layer management functionalities in the context of multi-layer networks. The Path Computation Element (PCE) [39] and the Virtual Network Topology Manager (VNTM) [84] address cross-layer path computation and cross-layer topology management, respectively. In this section, we will describe the basics and outline the reach of these management subsystems in the context of multi-layer network management.

### 2.4.1 Path Computation Element (PCE)

The Path Computation Element (PCE) [39] is a standardized solution for facilitating optimal constraint-based path computation in (G)MPLS networks. In this architecture,

a Path Computation Client (PCC) can request the computation of a path under specific constraints to the PCE, which in turn uses its Traffic Engineering Database (TED) to compute the requested paths. The communications between the PCC and the PCE have been standardized by the IETF in the form of the Path Computation Element Protocol (PCEP) [85]. Note that, providing a detailed description of the PCE is out of the scope of this thesis. Instead, our main focus is to position the PCE in the context of multi-layer settings. Readers are referred to [86, 87] for further details on this subject.

There are different choices for the implementation of the PCE, wherein a PCE server can be implemented within a Label Switched Router or as an external PCE that is implemented as a third-party subsystem. The use of a centralized server (i.e., an external PCE) is especially beneficial in networks requiring complex path computation such as Wavelength Switched Optical Networks (WSON). In these networks, the implementation of specific functions for path computation with high complexity in each switch can drive up equipment cost. On the other hand, centralized servers on dedicated hardware designed for path computation would be a more cost-effective solution, especially in large networks. A unique selling point of the PCE architecture is its ability to extend path computation capabilities across multiple domains, including multi-layer networks. PCEs preserve topology confidentiality, which is essential in commercial network scenarios. Also, the decoupling of path computation from network devices means that operators can employ third-party boxes and can control path computation mechanisms and policies with ease, even in a multi-vendor scenario.

To date, the PCE protocol has been standardized for use in MPLS networks and current standardization work is focusing on extending the protocol for supporting networks using the GMPLS control plane [88], and for wavelength assignment in WSON networks [89]. Extensions to the PCE protocol have also been proposed to compute optimal inter-domain paths on a fixed domain chain using the Backward Recursive Path Computation (BRPC) [90].

Current standardization work is also focusing on the use of PCE for multi-layer path computation [84]. PCE has positioned as the candidate solution to overcome the limitations of such networks for providing effective multi-layer Traffic Engineering (TE), which are primarily attributed to the lack of shared network resource knowledge between layers [91]. In this network scenario, multi-domain path computation is not a high priority, but multi-layer path computation is especially relevant as most large carriers typically run a transport as well as an IP/MPLS infrastructure, and would benefit significantly from automated multi-layer path computation. While no definitive solution for multi-layer path computation with the PCE is available today, [84] defines three different mechanisms for the same, namely (i) Single PCE Inter-Layer Path Computation, (ii) Multiple

PCE with Inter-Layer Path Computation and (iii) Multiple PCE without Inter-Layer Path Computation.

The Single PCE Inter-Layer Path Computation uses one PCE to compute paths between multiple layers. The PCE has the global knowledge of all topologies and can compute paths across different layers. The Multiple PCE with Inter-Layer PCE communication approach uses PCEs on each layer, having topological visibility restricted to their own layer. This model adapts to the Inter-Domain path computation scenario, where different PCEs are chained to compute a strict path from source to destination. In the case of PCE without Inter-Layer PCE communication, each PCE computes loose paths from ingress to egress LSP, in this case higher level LSP to lower level LSP, building the path traversing every PCE until the destination is reached—referring again to the PCE Protocol model of Inter-Domain path computation with loose hops. Out of these mechanisms, the single multi-layer PCE has the best performance, but the Multiple PCE with Inter-Layer PCE communication approach is better suited for most carrier-grade networks, given the administrative segmentation between the IP/MPLS and transport network in most carrier organizations. The implementation of the same is demonstrated in [92] where the authors use the existing standards to facilitate inter-layer path computation in a MPLS over WSON network scenario.

Aside from the already mentioned schemes for multi-layer path computation, the algorithmic issue for computing such paths is of utmost importance. The main challenge that multi-layer path computation algorithms face is combining different layer-specific constraints to find optimal or near-optimal cross-layer paths [93]. The majority of constraints that condition the set up of a lightpath in an optical network (e.g., wavelength continuity, attenuation, etc.) demand solutions that differ from traditional circuit-based computation methods. In light of this, algorithmic solutions in the context of multi-layer networks must take into account other variants, for example, network device capabilities for performing multi-switching and wavelength conversion in order to compute cross-layer paths under given constraints. In the literature, various multi-layer path computation algorithms have been proposed [94, 95, 93, 96, 97, 98, 99].

In [94, 95] B. Jabbari et. al. discuss on the constraints and possible solutions for computing traffic engineering paths in multi-layer switched networks. They propose a solution in which a network graph is transformed into a channel graph that explicitly exposes the constraints of nodes and links, which otherwise are not visible. Authors in [96] extend this approach by combining the transformation technique with a simple heuristic aimed to cope with the increased complexity of the new graph. They also introduce and evaluate a Constrained Breadth First Search (C-BSF) technique where

| Constraint Type | | Proposed Path Computation Algorithms |
|---|---|---|
| **Prunable** | *Based on bandwidth requirements*<br>*Based on protection requirements*<br>*Based on policy constraint requirements* | • Several variants of Constrained Shortest Path First (CSPF) [100] |
| **Non-Prunable** | *Additive (e.g., attenuation, dispersion, delay, etc.)* | • K Shortest Paths (KSP) [101, 102, 103] |
| | *Non-Additive (e.g., wavelength continuity, switching capability, etc.)* | • Common Vector [94]<br>• Constrained Breadth First Search (C-BFS) [96]<br>• Channel Graph-based solution [94, 95]<br>• Label-Layer Graph-based solution [96]<br>• Auxiliary Graphs [104, 105]<br>• Dynamic Virtual Network Topology (VNT) Configuration Algorithm [93] |

TABLE 2.1: Multi-layer Path Computation Algorithms.

constraints are evaluated on-a-fly fashion based on the BSF search algorithm. Authors in [97] developed an heuristic called *Min-phys-hop* routing and a wavelength assignment algorithm, which assigns a weight to each optical path that corresponds to the number of physical links that comprise it. Based on the belief that an efficient path goes over the minimum number of physical network devices, the least-cost path is chosen. One of the main limitations of this algorithm is that it does not consider network nodes capabilities such as Optical-Electrical-Optical (OEO) conversion or wavelength conversion as considered by other algorithms [94, 95].

In Table 2.1, we summarize the main approaches toward solving the algorithmic issue of multi-layer path computation. Note that, this table does not intend to provide an exhaustive study on multi-layer path computation algorithms. Instead, it provides an introductory and representative list of solid contributions in the field. We have categorized constraints into prunable and non-prunable classes as defined by authors in [94]. The prunable category refers to all those constraints that can be met by simply discarding the elements that do not comply with the required feature from the path computation process, e.g., bandwidth requirements, wherein potential paths not meeting a bandwidth constraint can be excluded from the path search process. The non-prunable category, on the other hand, usually comprises the set of constraints that require more complex computation strategies taking into account multiple network attributes along the whole path.

For instance, in networks where certain nodes are endowed with wavelength converters, the lack of a common wavelength along the entire path is not sufficient to discard the latter as a potential candidate, since we must also assess the wavelength conversion capabilities at intermediate nodes along the path. Indeed, determining whether a constraint is prunable or not is also subject to design requirements. Multi-constraint path computation algorithms generally follow one of two approaches, namely, computing paths over the raw network graph and then performing robust runtime constraint evaluation (e.g., algorithms based on linear programming), or graph transformation techniques, where network graphs are transformed into elaborated graphs capable of representing a set of given constraints [96]. For more details on the algorithmic issues we refer readers to the references found in Table 2.1.

Note that, despite of all the ongoing standardization efforts in the field of cooperative path computation—some already discussed herein—many issues still remain open for research. Consider for example, the computation of a cooperative path between several PCE's based on proprietary objective functions operating on non-standard constraints. Current framework does support encoding of standard and proprietary objective functions [106]. However, there are no available mechanisms in the current definition of PCEP for conveying vendor-specific information on which proprietary objective functions rely. Authors in [107] have recently proposed a mechanism for conveying vendor-specific constraints in PCEP in which they define a dedicated object—the "vendor information object"—to convey vendor-specific information. In light of this, there is still many to be done in the scope of multi-layer path computation. The realization of all these efforts and proposals are indeed required in the way for achieving a truly mature technology suitable to the complex requirements of multi-layer and multi-vendor carrier-grade infrastructures.

### 2.4.2 Virtual Network Topology Manager (VNTM)

The Virtual Network Topology (VNT) is the network topology formed by lower layer LSPs and the logical view of these connections in the upper layer [66]. The relationship between both layers is created with "virtual TE links" [108]. The virtual TE links are potential connections between two nodes at the upper layer, which are based on possible connections at the lower layer (i.e., not fully provisioned LSPs). The Virtual Network Topology Manager (VNTM) is an entity in charge of maintaining the topology of the upper layer by connections in the lower layer [84].

To optimize the overall use of network resources in multi-layer environments, there are basically two requirements: $(i)$ a cross-layer path computation strategy to compute

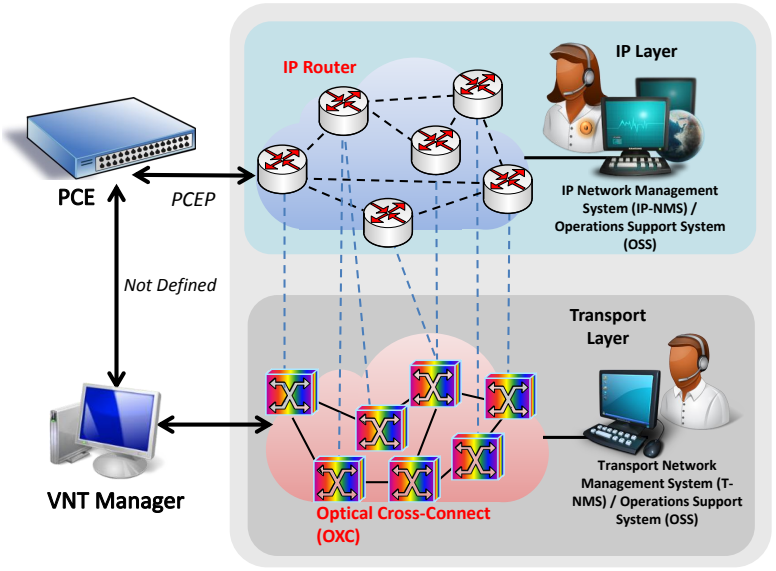


FIGURE 2.6: An example showing one possible PCE/VNTM configuration.

end-to-end inter-layer paths and (*ii*) mechanisms to control and manage the VNT by provisioning and releasing connections in the lower layer. In this regard, the VNTM and the PCE are both key entities for coordinating and managing multi-layer paths. While the PCE is responsible for computing the path between endpoints, the VNTM initiates the signaling for circuit setup or decommissioning in the transport layer [92].

There are two possible cooperation models for inter-layer path computation, namely, the *PCE-VNTM* or the *NMS-VNTM* cooperation models [84]. To illustrate the relationship between the PCE and the VNTM for multi-layer path computation in a PCE-VNTM model, let us consider the example shown in Fig. 2.6. In this case, a single PCE is used to compute paths between both layers (i.e., a single inter-layer path computation strategy, as described in the previous section). Let us assume that, the single-layer PCE fails to compute an inter-layer path because no logical connection is set up in the upper layer (IP; PCE). In such conditions, the PCE can request or suggest to the VNTM additional connections in the lower layer (OXC; VNTM). If the PCE has visibility of the lower layer topology it can explicitly suggest a given path or it can just exchange information on the upper layer request and wait for the advertisement of the new virtual TE link. Moreover, the VNTM could change the connections to the upper layer if its policies indicate that a better configuration exists. The operation mode of the VNTM could be any solution even using another PCE to compute lower layer LSPs.

Another approach to inter-layer path computation is when the VNTM is part of the NMS (NMS-VNTM model [84]) (cf., Fig. 2.7). This model assumes that the VNTM can be embedded within the NMS to cooperate in the set up of lower layer LSPs. In this scenario, the NMS requests the PCE to compute a path and upon receiving the result

of such computation, the NMS is able to request the VNTM to create a new connection. The interface to request a connection is not currently described or defined in any RFC (this is why we represent the interface between the NMS and the VNTM in Fig. 2.7 with an interrogation mark). In [84] a TMF standard interface is proposed, while authors in [92] propose the utilization of the PCEP protocol with a new message to suggest a new route to the VNTM, so the VNTM can decide, based on its policies, if creating the connection or not.

Based on the previous definition of the VNTM, the virtual TE link creation is done from the source IP/MPLS router to the destination router. This means that the VNTM must have information not only about the lower layer, but also about the interlayer connections between the IP/MPLS routers and the OXCs. However, automatically building and accurately keeping updated the inter-layer connections remains an open issue. State-of-the-art protocols, such as the Link Management Protocol (LMP) [109], offer palliative solutions. This point-to-point protocol, defined by the IETF, is used by GMPLS devices to feature link discovery, i.e., determine data plane connectivity of network nodes to and from its neighbors. Link verification and fault isolation are also featured by this protocol to check on the status of links and to isolate faults that may occur, respectively [110]. Nevertheless, LMP is not widely used by ISPs mainly because current deployed multi-layer networks do not provide full integration of GMPLS technology. Actually, in practice most ISPs remain relaying on databases, most of which require human intervention for feeding and updating inter-layer topological data.

In summary, the emergence of management subsystems was envisioned to simplify and partially fulfill a set of management functionalities that were not covered by existing

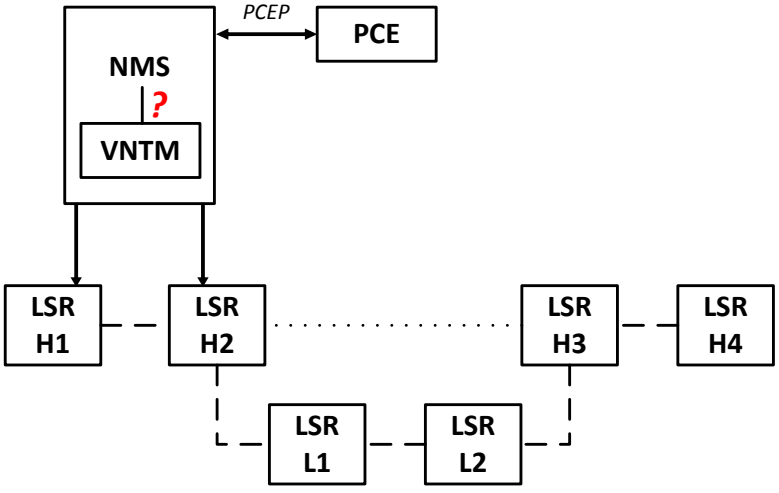


FIGURE 2.7: Another example of a PCE/VNTM configuration with a VNTM entity embedded within the NMS.

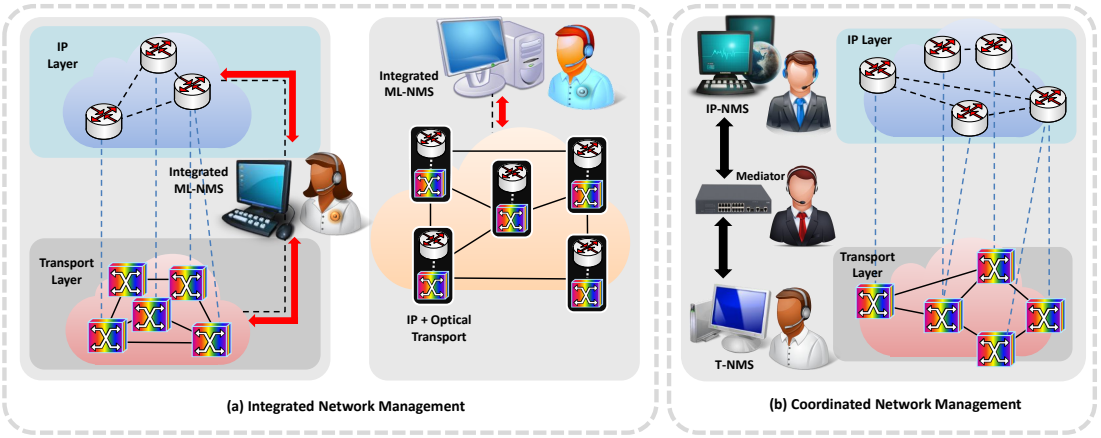


FIGURE 2.8: Integrated vs. coordinated approaches for multi-layer network management.

solutions. This makes mandatory to consider the integration and future requirements of third-party subsystems in multi-layer management infrastructures.

## 2.5 Future Trends in Multi-Layer Network Management

### 2.5.1 Coordination vs. Integration

Integrated and coordinated approaches both offer a path for addressing current multi-layer network management challenges. The integrated approach (cf. Fig. 2.8a) proposes to unify the two separate NMSs into a single entity, enabling management automation and reduction of OPEX. This figure shows two possible configurations. In the first one, the IP and Optical devices are part of separate layers—as in traditional multi-layer deployments—but their management planes are unified under a common NMS. In the second configuration, the IP and Optical equipment are integrated into a single box with a unified management plane. Examples of the former are the integrated control plane and management plane frameworks [64, 77, 78, 79], and for the latter, the case of Juniper's hybrid node [37].

Unfortunately, these approaches pose complex challenges both technically and operationally. For instance, multi-layer network management issues cannot be fully addressed by the integration of network control planes per-se, as they do not fulfill all management competencies. Therefore, under this scenario, the management planes of both layers would also require a level of integration; otherwise, cross-layer service management operations (i.e., provisioning, monitoring, alarm correlation, scheduling, etc.) will

not be possible. Furthermore, the integration of network management systems aggregates a measure of disruption. More specifically, the integration of the IP Network Management System (IP-NMS) and Transport Network Management System (T-NMS) entails a change of perspective in current carriers' practices, and therefore, a major investment must be done to acquire and adapt to the new management environment. The traditional separation between the IP and transport network layers has demonstrated significant resistance to such a game change, given its implications on the operational, functional and business strategies. Moreover, a successful network management integration can only be useful for a single domain, since domain administrators are reluctant to sharing performance information about their networks.

In light of all these limitations, coordinated network approaches (cf. Fig. 2.8b) are positioned as reliable solutions for multi-layer network management. Coordination of cross-layer network operations can help not only to reduce operational and capital expenses, but also to significantly simplify operational processes and reduce administrative burdens. A coordinated approach is based on the idea of a mediator system (shown in Fig. 2.8b) capable of overcoming the barriers of protocol differences and network equipment heterogeneity. A mediator can address the MLI and MVI issues without requiring major changes in network practices or the technologies currently being used by network operators on both layers. Additionally, the coordinated approach can easily be adapted to meet future application demands, as its adoption does not cause any disruption to current network practices.

One emerging initiative in this direction is the *ONE Adapter* [111], a solution for enabling interoperability in a coordinated fashion. The ONE adapter is based on a mediator model, which has been designed to operate between the IP/MPLS and the transport layer NMSs. It allows system automation and coordination of cross-layer network management functions. This initiative proposes a non-disruptive solution that only requires to interface with the IP-NMS and the T-NMS. Furthermore, it does not require changes ro current carriers' practices. The ONE adapter is particularly useful for management actions such as coordinated self-healing, coordinated IP traffic offloading, and coordinated service provisioning. The ONE approach also allows coordination of network management functions in a multi-vendor environment, thus addressing the MVI problem as well. One of the key features in ONE is its ontology-driven nature, which enables syntactic adaptations based on shared semantic knowledge of different vendor spaces.

Table 2.2 summarizes the advantages and drawbacks of coordinated versus integrated multi-layer network management.

| Approach | Advantages | Drawbacks |
|---|---|---|
| ***Integrated*** | • It reduces the management and operational costs.<br>• It centralizes management tasks (single point of operation).<br>• It provides overall view of the multi-layer network infrastructure. | • It requires big changes in the network structure and in current practices.<br>• It is hard to adapt to meet future application demands.<br>• It hardly deals with multi-vendor interoperability issues.<br>• Its adoption disrupts the networks' regular activities.<br>• It is difficult to deploy.<br>• It requires to be integrated in all layers.<br>• It has high deployment costs. |
| ***Coordinated*** | • It requires neither big changes in the network structure nor in current operational practices and business workflows.<br>• It can be easily adapted to any type of environment to meet the future application demand.<br>• It can easily deal with the multi-vendor interoperability issues.<br>• Its adaption does not disrupt the networks regular activities.<br>• Its deployment is easy and needs only to be interfaced with the IP and the transport NMSs.<br>• It allows functionalities, which do not exist in the control plane frameworks.<br>• Its functions are domain and layer independent. | • Cannot directly manage the networks nor it can make changes to the IP layer or the transport layer without the intervention of the NMSs at each layer.<br>• Its normal workings depend on the responsiveness of third party systems (e.g., PCE) as well as the IP and the transport layer NMS. |

TABLE 2.2: Advantages and drawbacks of integrated vs. coordinated multi-layer network management solutions.

### 2.5.2 Enabling Technologies for Coordinated Approaches

In addition to the numerous advantages and benefits that a coordinated approach can provide to address the MLI and MVI problems in the context of network management, we have identified a number of emerging technologies that can assist in the implementation of truly coordinated platforms.

Beyond the scope of traditional mechanisms for network management, there are a number of technologies capable of enabling enhanced functionalities to this critical area of networking. MTOSI [46], NETCONF [10], semantic approaches and OpenFlow [112] are among the most relevant enablers for providing support to the field of multi-layer

network management. Despite the fact that some of these technologies (e.g., semantic technologies and OpenFlow) were not initially designed for the purposes of network management, they have an interesting potential that can ease and address the management interoperability gap in multi-layer and multi-vendor settings. In this section, we will provide insights on the potential integration of these technologies in order to understand what makes them appealing for their applicability in the area of coordinated Multi-Layer Network Management (MLNM). It is worth highlighting that given the impact of OpenFlow, this technology will be further analyzed in Section 2.5.3 as part of the Software Defined Networking (SDN) umbrella.

*NETCONF*

To fill the existing gap in the configuration of IP network equipment, in early 2003 the IETF created a working group to define and develop a *standard* configuration protocol. The result of this effort is the Network Configuration Protocol (NETCONF), specified in [10] as an IETF standard. NETCONF was defined to cope with the needs of providing configuration state maintenance, transactional-safe operations across multiple devices, separation of configuration from operational data, concurrency, consistency, and support to multiple configurations, in a standard and easy way of use. This set of requirements gather the most remarkable characteristics of CLI-based mechanisms along with the desired features of network providers in the scope of configuration management [113].

Some of the key features of NETCONF are: (*i*) the ability to distinguish configuration data from operational data, i.e., variables that can be set by the administrator from statistics, alarms, notifications, etc.; *(ii)* support to transactions, meaning that, it ensures completion of configuration tasks—not only on a device basis but even, on a network basis—otherwise, rollback operations are automatically performed; (*iii*) transport protocol independence; (*iv*) support to configuration locking; and (*v*) filtering mechanisms that enable selected retrieval of configuration data [12]. Most importantly, NETCONF features automated ordering of operations, which means that the complexity of task sequence ordering is pushed from the operator's side into the device. Its design is flexible and extensible enough to be implemented and deployed by all vendors. Note that this is not an advantage over other protocols, such as SNMP, but is a *must* if NETCONF intends to become a widely adopted standard.

The complexities of SNMP and the proprietary nature of CLI-based mechanisms, have

pushed toward the initial implementation of NETCONF in IP/MPLS vendor's equipment (e.g., Juniper and Cisco) as a means to guarantee the performance of the configuration procedure. The inclusion of NETCONF will enable operators to deal in a standard and reliable way with multi-vendor infrastructures, significantly decreasing OPEX. However, as exposed earlier in Section 2.1.2, the definition of network management information within NETCONF raises clear interoperability issues that still represent an important limitation to its wide acceptance and deployment. Most recently, YANG [13]—the IETF's proposal for a standard data model—has become the strongest candidate for the formal representation of network configuration management information. It provides well-defined abstractions of the network resources that can be configured or manipulated by a network administrator, including both devices and services. Currently, the IETF is working on the definition of standard YANG modules, to which vendors are expected to comply with in the future. To date, several internet-drafts have been introduced for the definition of the interface, IP, routing, SNMP and system management data models [114]. However, there is still a long path before NETCONF and YANG become mature technologies and established as the default standard for IP network management configuration. Though, the future of NETCONF and YANG is promising and the interest of network device vendors is growing, almost eight years after its initial proposal, CLI-based mechanisms continue to be the preferred way for configuring network devices. For this reason, and based on the fact that a solution is required for current network settings, we envision semantic technologies as a promising enabler for the integration of dissimilar protocols in the scope of network management. We proceed to briefly describe the potential of these technologies in the scope of a coordinated platform dealing with heterogeneous protocols.

*Semantic Adaptations*

Semantic adaptations serve as a promising approach for enabling future coordination of multi-layer management. When combined with current available technologies (e.g., ontologies, data mining, artificial intelligence, natural language processing, etc.), semantic approaches can provide the means for achieving true interoperability in the configuration management of multi-vendor environments.

The benefits that semantic approaches can provide go beyond the scope of seamless device configuration, and can be devised for any other scope in which the existence of diverse protocols or mechanisms lead to heterogeneous scenarios. They provide an evolutionary solution to current MVI problems through an intelligent, flexible, and extensible approach. Semantic adaptations can thus improve and enhance the experience

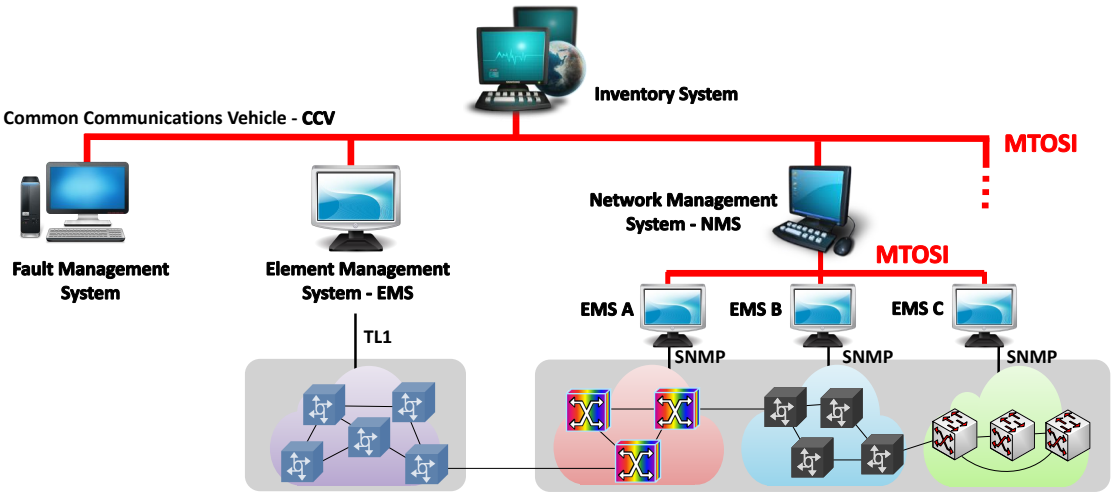


FIGURE 2.9: MTOSI reference architecture example (adapted from "Framework DDP BA TMF518 FMW Version 1.2" [1]).

of the final user, abstracting network managers from the particularities of each language or format, thereby removing the complexity that is left in the hands of operators (which is error-prone and highly restricts the level of automation of management tasks).

Semantic technologies can be combined with machine learning techniques [115] to develop software agents capable of semantically and syntactically adapting configurations for easing tasks in multi-layer and multi-vendor environments. In the literature, several research efforts can be found aligned with the use of semantic technologies to address the problems inherent to the network management domain [116, 117, 118, 119]. For example, in [117, 118] H. Xui and D. Xiao consider the application of ontology languages to bring intelligence into the network management plane to enable automated configuration. Authors propose a general model in which three ontology-related languages are considered. They propose, firstly, the Ontology Web Language (OWL) [54] for modeling the domain knowledge, secondly, the Semantic Web Rule Language (SWRL) [120] to add behavior information—such as axioms and constraints—and finally, OWL-S [121]—a semantic markup language for Web Services—to automate the execution logic based on the use of Web-Services. In short, authors envision the use of Semantic Web Services for automation of network management and propose standardization of network management information on this basis. Moreover, in the work led by J. López et al. in [119], authors thoroughly analyze and compare several management information definition languages on the basis of their semantic expressiveness. The goal is to integrate different management models based on ontological mapping and merging techniques to create a global management ontology encompassing a huge set of information models (e.g., Structure of Management Information version 2 - SMIv2, Managed Object Format / Common Information Model - MOF/CIM, etc.). Finally, A. K. Y. Wong et al. [116]

proposed a generic ontology-driven solution to the interoperability problem in network management. In this approach, ontology concepts are matched based on the similarity measure obtained through a novel function developed by the authors.

As seen, semantic technologies have already been considered in previous research studies to address the interoperability issues in network management as well as in many other fields (e.g., health care, product design, biomedicine, etc.). Nevertheless, there are still many open research lines that must be explored in order to exploit and properly use the resources offered by device manufacturers.

*MTOSI*

In the field of optical transport networks, MTOSI [46] has emerged with great force as a technology for enabling standard communication between multiple OSS (e.g., Element Management Systems, Network Management Systems, Service Activation Systems, Fault Management Systems, etc.), covering both service and resource management. This standard has been specified by the TeleManagement Forum (TMF) to provide a common interface to manage *multi-technology* networks. It is an XML/Web Service-based interface, that is independent from the underlying transport technology. MTOSI enables interoperability between the Service, Network and Element Management Layers of the Telecommunications Management Network (TMN) layering model.

The general specifications of the interface maintain that MTOSI will be used between OSSs within the same administration [1]. Hence, it covers the interoperability requirements of current Service Providers which use multiple management systems to manage complex multi-vendor and multi-technology networks.

To illustrate the MTOSI reference architecture let us consider the example shown in Fig. 2.9. This figure provides a view of the communication between different OSSs enabled through MTOSI at different levels. The interaction between the management systems is done over the Common Communication Vehicle (CCV). As depicted in the figure, in typical network settings (i.e., a multi-vendor scenario) a NMS—or other type of OSS—directly connected to the CCV can be at the same time managing underlying Element Management Systems (EMSs), which in turn manage network elements of a single vendor based on other management protocols (e.g., SNMP, TL1, CMIP, etc.). For this example, a Fault Management System providing an MTOSI-based interface can retrieve inventory from an Inventory OS to effectively fulfill its management functions.

During the last few years, MTOSI seemed to become the future standard for enabling interoperability between OSSs in the transport layer. This web-based interface was positioned as a rather appealing technology for overcoming the limitations in the scope of network management. However, a change in perspective in the field has made MTOSI lose momentum and instead, attention has turned most recently to Software Defined Networking (SDN), a new emerging technology with the power for taking the management solution to the next level.

SDN has become an increasingly popular concept whose potential certainly opens the field for research and innovation. For this reason, in the next section we will provide insights on SDNs capabilities and on the challenges from a management point of view.

### 2.5.3 Software Defined Networks (SDN)

We could say that Software Defined Networking (SDN) is the current fulfillment of an old-time promise: providing the possibility of programmability of network functionalities, while offering a clear path for network management to follow the same direction of other Information Technology (IT) fields toward virtualization.

The idea of a flexible real-time control of network functions has been considered several times in the past, with proposals ranging from the use of "programming packets" to transmit the desired behavior to network boxes, up to the implementation of adaptive control both in software and hardware. However, the combination of radical decoupling and open interfaces that constitutes the kernel of SDN is a novel proposal that has gained a strong momentum, especially, with the advent of a protocol that demonstrated the feasibility of this approach, and allowed the deployment of real-scale SDN-based networks: OpenFlow [112].

The complexity and lack of flexibility of standard network devices has made network experimentation and innovation highly difficult at all scales for academic researchers. Any change to the software embedded in each device had to be coordinated between vendors in order to make the distributed control algorithms interoperable. Therefore, evolving at the pace required by research and experimentation was extremely difficult. In this context, OpenFlow was born as a cornerstone. The first step was to develop the ability to program switches from a remote controller. Realizing that this implied external software-based control of the data plane, bypassing traditional L2 and L3 protocols and associated configurations, was a natural consequence.

Software Defined Networking relies on two main assumptions. The first is a radical separation between the control and the data planes, located in two (most often physically)

independent entities: the controller, in charge of the control plane, and the switches, responsible for the functions in the data plane. The choice of singular and plural in the definition above is completely intentional: although not required by the model, the obvious deployment consists of a single controller taking care of several switches in a certain realm. The second assumption is the availability of an open protocol between controllers and switches, allowing for a free combination of elements from different vendors to provide network functions, and of an open interface to the control plane, so the controller can be uniformly accessed by other components participating in the network, such as sources of network intelligence or applications in general.

The most widely deployed SDN protocol, OpenFlow, is based on the definition of rules from the controller to be applied by the switches when receiving packets. Rules are fired by matching certain parts of the packets (or the path they arrive through), and contain actions to be applied to those packets, such as forwarding them to a certain path, making some changes to them, or even discarding them.

In summary, in SDN control decisions are taken by a central element, while switching decisions are actually applied by distributed elements. A common protocol allows the controller to communicate its decisions to the switches. Having this central element translates into the possibility of abstracting the network into a single element, as it becomes the one in charge of the whole network behavior. Furthermore, the common protocol acts in a similar way to a processor instruction set controlling its registries, processing units and peripherals, and therefore the network becomes a programmable entity, suitable to be controlled in the same way as any other element in the whole computing infrastructure.

A general architecture for SDN-based network management is proposed in [122] (shown in Fig. 2.10), where an *SDN Mediator* communicates with applications and services, and translates requests to the physical network components. This mediator relies on a database containing network topology and component information, and it is able to provide a fully virtualized view to applications and services. The mediator controls several processes:

- ***Discovery***. It enables the SDN users to discover and register to the Service Mediator. As a part of the discovery process, the SDN users may negotiate capabilities with the service mediator.
- ***Provisioning***. It allows the Service Mediator to provision the underlying network resources. While in principle the provisioning process should rely on OpenFlow, it is conceivable the use of other protocols to create or adjust traffic engineering connections.

- ***Monitoring***. This allows the Service Mediator to interface with the underlying network to gather topology information at an abstract level, and detect the network failures that may impact applications and services.

A more radical approach is taken by a group of the original OpenFlow proponents [123], who present a unified control for packet and transport networks, claiming that with separation of data and control, and the treatment of packets as flows, together with the introduction of circuit-flow features in the OpenFlow protocol, a unified architecture becomes realizable for converged packet-circuit networks. OpenFlow abstracts each data-plane switch as a flow table. It allows the definition of a flow to be any combination of L2-L4 packet headers for packet flows, as well as L0-L1 circuit parameters for circuit flows.

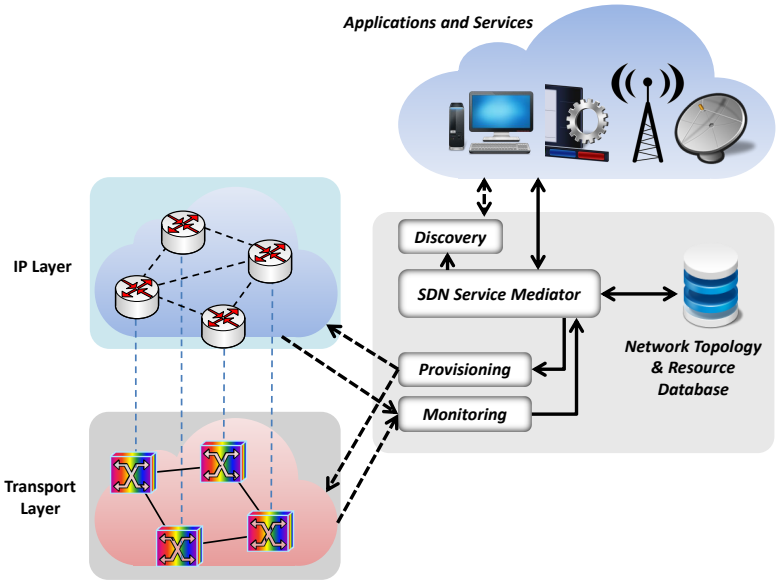


FIGURE 2.10: Mediator architecture for SDN-based management.

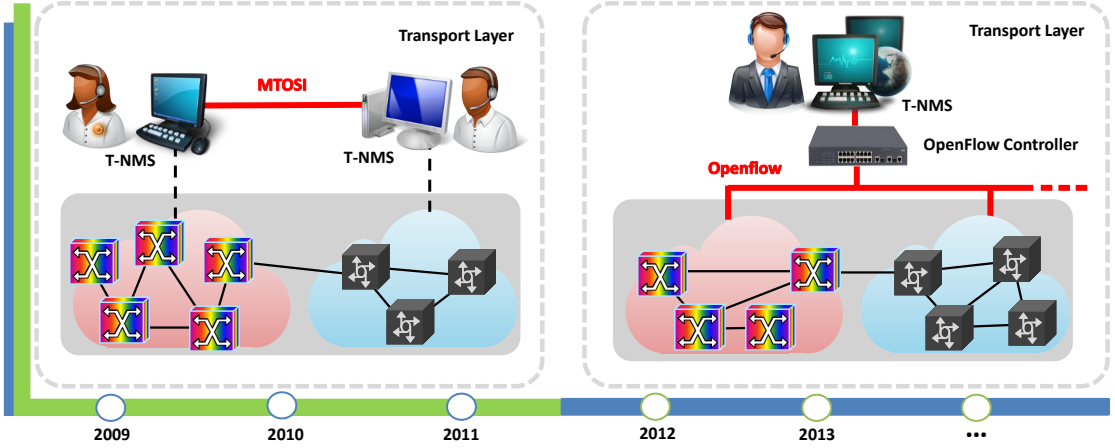


FIGURE 2.11: Network management trend: SDN over MTOSI.

Whether a mediator-based or a SDN-only approach is followed, it is important to remark four salient characteristics implied by the use of SDN for multi-layer network management:

- Since the network elements are controlled via a (few) uniform protocol(s), the management database can be greatly simplified.
- Management commands and/or configuration are always translated into pairs of the type `(device,rule)`, and it is possible to override layer separations, device original functions, and other potential limitations in other models. A multi-layer management plane is a natural consequence of applying SDN, even if not a full integration is pursued.
- The northbound Application Programming Interface (API) can be adapted to provide different management views to the application services, not limited by individual elements, links or layers being managed. The network functions can be fully virtualized.
- This virtualization possibilities translate into an easy extensibility of the configuration programmatic interface, able to being updated according to the application/service needs, even in a real-time or ad-hoc manner.

Another important aspect is that with the advent of SDNs, the range of multi-layer capabilities has significantly increased, since open programmability enables that new applications can take control of network resources across all layers in the OSI stack. Observe that, we have focused on multi-layer capabilities across the transport and network layers of a typical carrier-grade network, but SDNs also bring a plethora of possibilities to upper layers, allowing applications to intelligently combine network functions at different layers. Consider for instance some of the recent research advances combining the utilization of OpenFlow and the Multi-Path Transport Control Protocol (MPTCP). In this particular example, the goal is to intelligently and transparently assign traffic to multiple paths in order to improve the resilience, the stability, and the performance of different types of communications [124, 125, 126, 127, 128]. Another interesting approach can be found in [129], where a cross-layer cooperation module is defined between MPTCP and the Locator/Identifier Separation Protocol (LISP). In this example, the goal of the authors is to improve both the performance and the transfer time between the endpoints.

An example involving even a higher number of layers can be found in [130], wherein the authors propose a methodology to extend Open VSwitch (OVS)—an open-source OpenFlow switch—to support L4-L7 service awareness. Moreover, SDN for transport

networks [131] [132]—in correspondence to our earlier discussions—has recently gained a lot of attention, and is considered one of the potential candidates for enabling packet/optical integration, and thereby, improve existing multi-layer path provisioning techniques [133, 134, 135]. Indeed, there are ongoing efforts within the Open Networking Foundation (ONF), for standardizing transport extensions to OpenFlow so as to enable SDN in optical networks. In the meanwhile, several approaches to SDN for the transport layer have already been developed. For example, ADVA, in a joint effort with IBM and the Marist College, have recently demonstrated an SDN solution for transport networks, mainly targeting the dynamic set up and tear down of wavelengths between data centers. Overall, SDN has become a revolutionary paradigm shift in the telecom field, and it is expected to have a significant impact on our conception of networking. The potential is remarkable, but the status of SDN-based applications enabling multi-layer functions in carrier-grade networks is still in a very early stage of development, so substantial research is needed before we can start witnessing the deployment of solutions in operational networks.

It is unquestionable that the advent of SDNs has weakened the power of MTOSI, which, only a few years ago, was positioned as the predominant trend for enabling transport network management interoperability. The paradigmatic differences between the two approaches has shifted the tendency toward SDNs. As shown in Fig. 2.11, this shift suggests that, in the future, the interest will no longer be focused on the communication between multiple OSS (i.e., the MTOSI paradigm), but instead, the new paradigm will be the one-to-multiple approach followed by SDNs for achieving true network programmability and virtualization through open and clear APIs.

## 2.6 Summary of Lessons Learned

The multi-layer core infrastructures of large ISPs have evolved as administratively separated ecosystems. This business-driven separation has led to the operational and managerial isolation of both layers. The lack of mechanisms enabling communication from a management perspective has clearly derived in interoperability issues between layers. Several initiatives can be found in the fields of control plane and data plane technologies, though none of these is capable of addressing the needs and requirements that arise from the management point of view. In addition to this, the emergence of new trends in the field of IP over Optical transport (e.g., hybrid nodes, proprietary multi-layer NMSs, etc.) pose new challenges in the management of heterogeneous network environments.

The integration and coordination of network management solutions in the context of multi-layer networks are among the most predominant approaches for overcoming the

isolation between the management ecosystems. These approaches can enable inter-layer interoperability, which can in turn significantly reduce the operational and capital expenses while facilitating a number of complex orchestrations required for management operations. Our research has identified that coordinated approaches seem to be the most suitable alternative, especially, for achieving automation of cross-layer operations, avoiding the redundancy of management functions, reducing operational and capital expenses, providing a smooth evolutionary practice, while offering higher flexibility. Examples such as the ONE Adapter in [111], as well as the mediation scheme for SDN-based management presented in Fig. 2.10, are indicative initiatives that envision the direction of a mediation model for truly accomplishing interoperability through the coordination of both—IP and Optical—layers.

In Table 2.3 we summarize the obstacles and pitfalls, the paths toward solutions, and the lessons learned in the management field of multi-layer networks. The aim is to provide a closer look into the main challenges, and summarize the issues that must be addressed by the research community in terms of future multi-layer management solutions. The future of network management for multi-layer network settings is not a clear panorama. However, the emergence of NETCONF, MTOSI, SDN supported through OpenFlow, as well as the consolidation of Web Services and many other new trends in the field of IP over Optical, are key indicators that a change of perspective is required to address the isolation of management ecosystems and improve multi-layer performance.

| Obstacles | Paths toward Solutions | Lessons Learned |
|---|---|---|
| *Absence of standards and/or broadly accepted mechanisms to enable network management interoperability* | • An early solution to the interoperability problem in network management was SNMP [44]. Despite being the de-facto protocol for network monitoring in IP-based networks, SNMP has failed to fulfill other scopes of network management, such as device configuration.<br><br>• Most recently, NETCONF [10] is envisioned for device configuration and MTOSI [46] for OSS interoperability in transport networks.<br><br>• OpenFlow has arised in the context of SDNs [112]. | The absence of standard protocols prevents automation of cross-layer management operations, leading to manual management of current multi-layer infrastructures. Standard protocols for network monitoring (e.g., SNMP and most recently Web Services) and network device configuration (e.g., NETCONF or OpenFlow) are not sufficient to enable management interoperability across multi-layer platforms. For instance, Network Managers require means (i.e., mechanisms, platforms) to coordinate between both layers to optimize resource usage, avoid duplication of network functions and automate/execute cross-layer workflows. |
| *Poor Coordination across layers* | • In-house developments are early attempts to coordinate multi-layer tasks in a static (i.e., pre-configured) way.<br><br>• Most recently, the ONE Adapter [111] has approached multi-layer management in a coordinated fashion.<br><br>• Coordinated systems based on SDN's (mediator models). | The lack of coordination between layers in multi-layer networks has led to duplication of network functions and long provisioning timescales. In-house developments have been for long time a way to flatten the issues in multi-layer management. However, the lack of flexibility makes them useful in scenarios where fixed solutions are sufficient. However, multi-layer management requires much more flexible, dynamic, programmable (i.e., configurable) solutions capable of overcoming the barriers of layer segmentation. |
| *Lack of Cross-Layer Network Management Automation* | • For instance, GMPLS control plane [57]. | GMPLS is positioned as the unified control plane solution for dynamic path provisioning in multi-layer networks. However, neither these solutions are widely deployed nor control plane technologies are capable of addressing all the needs and requirements from a management perspective (e.g., proactive execution of policy-based workflows with cross-layer components, IP device configuration, etc.). |

| Obstacles | Paths toward Solutions | Lessons Learned |
|---|---|---|
| *Limitations for Proactive Enforcement of Policy-based Management in Multi-Vendor settings* | • In-house developments (i.e., umbrella NMSs). | Network programmability is a must in order to develop scalable solutions capable of adapting to the changing nature of network behavior. Network administrators require of flexible solutions that enable on-the-fly policy enforcement to bring dynamics to the multi-layer environment. |
| *Inadequate discovery mechanisms of Inter-layer connections* | • Manual Topological Databases. | Manual topological databases are error-prone, hard to maintain and present poor scalability features. Inter-layer discovery remains an open research challenge and requires of **automated** solutions capable of overcoming the restricted view (i.e., shared information) between layers. |
| *Deficient mechanisms for seamless network device configuration (e.g., the preferred configuration mechanism in IP-based networks is CLI)* | • NETCONF [10], a standard-based solution to the configuration issue in IP-based networks.<br>• OpenFlow [112]. | Standardization of network management protocols is not sufficient to overcome the configuration heterogeneity issue in multi-vendor environments. In this view, standardization of data models is equally relevant to the configuration domain to comply with a standard view of the network elements. |
| *Overburden of multi-layer functions which add on the basis of their complexity (e.g., computation complexity) - (embedded complexity)* | • Outsourcing of multi-layer functions, examples are the Path Computation Element (PCE) [39] and the Virtual Network Topology Manager (VNTM) [66].<br>• Virtualization and Network Functions Virtualization (NFV) [136]. | The advent of third-party (i.e., external) management subsystems, represent a unique opportunity to be integrated into future solutions to help solving cross-layer issues. For example, a PCE for outsourcing computation of multi-layer paths. In this field, yet some issues still remain open. For instance, the communication protocol between entities for coordination of cross-layer functions (e.g., the communication between the NMS and the VNTM in a cooperative model for multi-layer path computation has not been yet defined). |
| *Organizational Barriers (i.e., Department segmentation)* | • Integrated Network Management Solutions, e.g., Cyan's CyMS [64] or Juniper's PTX hybrid node [137]. | Integrated solutions have important implications on current practices. On the one hand, the traditional separation between the IP and Optical Departments is reluctant to a game change from the operational, functional and business perspectives. To this end, emerging solutions should seek for non-disruptive approaches from the business model point of view. On the other hand, current integrated solutions are subject to single vendor scenarios, a non-desired feature by network operators. |

| Obstacles | Paths toward Solutions | Lessons Learned |
|---|---|---|
| *Segmentation of Standardization Bodies* | • For instance, MPLS-TP [72]. | Solutions to multi-layer issues require of standardization bodies behind each domain (i.e., IP and Optical) to be aligned and to develop joint efforts to generate fully-compliant requirements and solutions to both technology layers. An example to this, is the MPLS-TP technology which begun as an ITU-T effort under the name of T-MPLS. However, IETF—the developers of MPLS standards—determined a set of inconsistencies between T-MPLS and native MPLS. They requested to extend the IETF's MPLS technologies to packet transport networks through the IETF Standards Process in a joint effort between both parties to consolidate a solution aligned to the IETF standards and fulfilling ITU-T requirements. |
| *High Operational and Capital Expenditures for managing Multi-Layer Settings* | • Network Functions Virtualization (NFV) [136] and Software Defined Networking (SDN) [122].<br><br>• IPoDWDM solutions integrate transponders directly into the IP routers enabling IP equipment to transmit ITU-compatible coloured wavelengths directly to the optical gear. | IPoDWDM solutions claim to reduce costs generously due to simplification of the network, at the cost of moving lower layer complexities to the upper layer, since routers have never had to deal with wavelength issues. This is somehow debatable as an assertion of this type depends on many factors (e.g., is implementation-dependent). Anyway, newly emerging solutions require to lower OPEX and CAPEX while maintaining the simplicity of network operation. Moreover, featured virtualization functions are undoubtedly of huge interest for ISPs, and will potentially contribute to significantly lower costs (CAPEX), while the benefits that SDNs technologies bring along to the field of network management will impact on the OPEX. |

TABLE 2.3: Obstacles and pitfalls, paths toward solutions, and lessons learned for managing multi-layer and multi-vendor settings.

# Chapter 3

# Internet Addressing

*This chapter aims to provide insights on the most prominent issues of the current Internet Addressing architecture and briefly introduce existing proposals on ID/Locator Split Architectures (ILSAs).*

## 3.1 IP-Based Addressing Scheme

Since the early days of networking, IP has been the absolute protocol supporting both routing and addressing on the Internet. Regardless of the well-known limitations of the current IP-based addressing scheme, the research community has put more attention in routing, with special focus in the inter-domain area rather than addressing [138] [139]. This research trend has actively promoted routing on top of addressing even with the awareness of the exhaustion of the IP addressing space [140]. Next, we will review the limitations of the current IP-based addressing scheme.

### 3.1.1 Limitations

The current IP-based addressing scheme carries along a number of limitations which significantly hinder the deployment of new applications and services in the Internet. These limitations are mainly related to *(i)* the depletion and *(ii)* the semantic overload of IP addresses. The former is mainly motivated by a design limitation and refers to the availability of the addressing space—taking into consideration the fact that the current IPv4 address space has nearly reached its limits [140]. The later is motivated by the lack of decoupling of location and identification, i.e., the two-fold functionality of IP addresses which clearly imposes a burden on the current Internet routing system. Accordingly,

these issues have an important impact over *(i)* multihoming, *(ii)* traffic engineering, *(iii)* mobility and *(iv)* resilient communications.

Multihoming is a common practice which significantly boosts the geometrical growth of the routing tables. This strategy is based on the notion of connecting a client (i.e., a computer or device) to more than one network, enabling fault tolerance capabilities and load balancing. In order to support multihoming, a site (i.e., an Autonomous System) obtains a Provider-Independent (PI) or Provider-Aggregatable (PA) prefix from its ISP, and further announces them through all its providers. PI and PA prefixes are blocks of IP addresses assigned by a Regional Internet Registry to a site. The main difference among them is that PA prefixes—unlike PI's—cannot be reused in the case that a site changes ISP. A multihoming site using PI addresses allocates its prefixes in the forwarding and routing tables of each of its providers. Therefore, PI prefixes are not aggregated. In the case of PA prefixes, the Internet Provider of a site could aggregate the customer (site) advertisement into a shorter prefix, when advertising the prefix to other customers or peers. In practice, ISPs have to advertise less aggregated IP routing prefixes to the Internet and rely on traditional and problematic longest-prefix match route selection algorithm of BGP [138].

Regarding mobility, as users demand connectivity on the move, addresses change. This change of address leads to significant degradation of the communication's quality or in the worst scenario service disruption. Notice that, even if an user changes of location, his identity remains the same. hence, while changes on the user location should be only reported to the routing layer, nowadays represent a change in the overall IP address. Despite efforts like Mobile IP—which aim to enable user mobility—the semantic overload remains an open issue. Moreover, resilient communications are also affected by the dual semantics of IP addresses. Consider for instance, a data center with a 1:1 protection scheme—with a set of primary and backup servers in different locations. In the case of failure of a primary server, the network layer will shift all traffic router to the failed server toward the backup server in place. This shift can certainly cause connection disruption.

In an effort to overcome the issues stemmed from the current addressing scheme, the networking research community is currently exploring two different lines of work. The first represented by the path drawn by IPv6 and the second clean-slate architectures for network addressing.

On the one hand, IPv6 has emerged as an evolutionary solution to the exhaustion of IP addresses. Nevertheless, the adoption and deployment of this technology is hindered by, *(i)* the operational expenses derived from migrating from IPv4 to IPv6—which requires of serious operational training of the networking staff in addition to firmware upgrades;

and (ii) the potential disruptions of network services—which can significantly affect regular ongoing network operation. Furthermore, IPv6 carries with the same issues of semantic overloading.

On the other hand, disruptive approaches to the current Internet Addressing scheme—such as that introduced in [31]—offer novel architectures explicitly designed to avoid the issues around IP addresses. Certainly, the risk of clean-slate solutions is related to the deployment of new technologies on top of an operational network–which will likely not allow compromising network connectivity. In light of this, a new line of research has emerged. This new line pursuits a new reference model, the so-called ID/Locator Split Architectures (ILSAs) aimed to decouple the semantics of IP addresses. A solution of this nature brought together with IPv6—or any other addressing scheme targeting the depletion of addresses—could certainly be a suitable solution to the current Internet Addressing problems. Next, we will provide a brief review of the concept of ILSAs and introduce a taxonomy for ILSA schemes.

## 3.2 ID/Locator Split Architectures (ILSA)

The ID/Locator Split Architectures (ILSAs) paradigm has recently emerged as a reference model aimed to overcome the issues of the current Internet addressing scheme. The "ID/LOC" philosophy—earlier introduced by Chiappa in [141]—consists in decoupling the dual semantics of addresses, i.e., separating host location from identification. Under this new paradigm, identifiers are confined to the application layer, and locators to the network layer.

Regarding the space of identifiers, its design faces two main challenges, namely, the identifier's lifetime and format. On the one hand, the lifetime of an identifier [142] basically impacts the number of signaling messages required to update the ID-to-LOC mapping tables. The shorter the lifetime of an identifier, the greater the accuracy of the mapping information, but the larger the number of signaling messages. On the other hand, the format of an identifier can be either *flat* (i.e., primitive) or *partitioned* (i.e., descriptive) [143]. The former refers to identifiers lacking of semantics, i.e., no information can be inferred from its structure (e.g., UUIDs Universally Unique Identifiers [144]). On the contrary, partitioned identifiers do have a semantic structure and thus, are user-friendly (e.g., URLs Uniform Resource Locators [145]). With respect to the space of locators, the challenges are related to bind a location entity to both, network topology and support for topological aggregation in an effort to ease routing performance.

Moreover, another relevant aspect in the design of an ILSA scheme is the mapping system. The fact that a single entity requires both, an identifier and a locator, translates into the imminent need of a system capable of mapping between IDs and LOCs. In the literature, several mapping systems have been proposed, however, they bring along pros and cons, which directly impact the overall performance of an ILSA scheme.

Figure 3.1 introduces our proposed taxonomy for ILSA schemes. This classification shows the ID/LOC generation challenge for three set of schemes, namely, Network-based and Host-based, in addition to, the Control Plane Mapping System. Overall, *network-based* schemes refer to those operating at the network level—typically on border routers, hence, no changes are required on the host level (i.e., end-nodes). LISP [20], Six/One [16] and GSE [19] are the most common and well-known schemes under this category. Network-based schemes can be further classified into, Map-Encap and Address-Rewriting schemes (cf., Fig. 3.1). The former category is based on the principle of tunneling—wherein packets are encapsulated to its destination. The later represents those schemes which operate in a similar way to the Network Address Translation (NAT) replacing a packet ID by a locator.

Host-based schemes refer to those strictly operating at the host level (i.e., on the end-points)—hence, no modifications are required at the network level. The main advantage of host-based schemes is that no costs of investment are required in the network. However, software updates are required—which is typically not attractive for software providers. HIP [17] and SHIM6 [15] are the best known examples of host-based schemes currently found in the literature. Another relevant conceptual difference among both schemes is that network-based are confined to a unique ID/LOC space, while host-based schemes have no restrictions on the uniqueness of the LOC or ID spaces. This feature

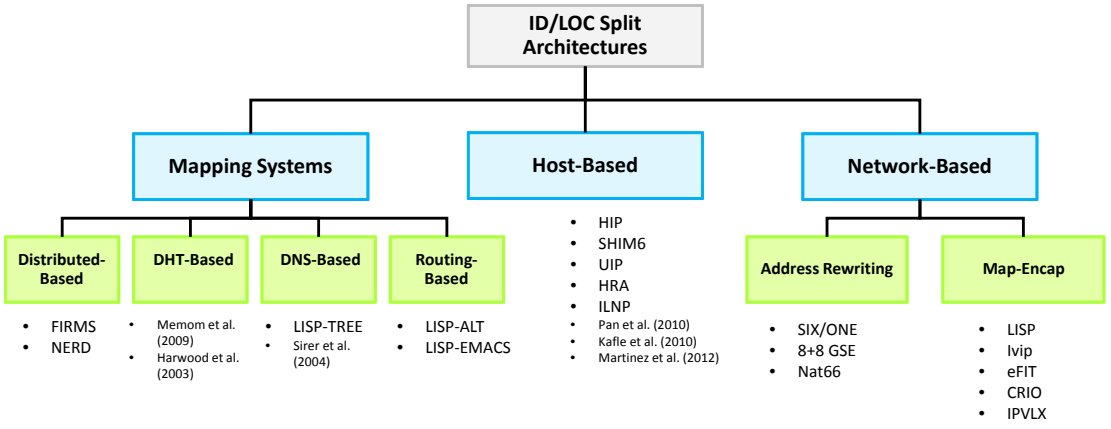


FIGURE 3.1: Taxonomy of ILSA schemes.

could be further exploited for scenarios wherein a two-fold location space could be convenient, for instance for global and local routing: or the case of a two-fold identification space, one for virtual objects and another for physical objects.

As mentioned earlier, when it comes to the design of an ILSA scheme another high-level challenge is the mapping system, i.e., the bidirectional mapping between IDs and LOCs. In the case of host-based schemes a different level of mapping may be required, for instance, between the various identification or location schemes in place. Furthermore, some schemes handle the mapping system over the data plane, whereas most recent trend is to push the mapping system to the control plane architecture—i.e., completely decoupled from the data plane. The mapping systems are conceptually supported by different technological approaches, namely, Distributed Hash Tables (DHT), Domain Name System (DNS) or routing protocols. The preferred approach is of utter importance in the design of the mapping system as it will tend to inherit the problems of the underlying technology.

## 3.3 Locator/Identifier Separation Protocol (LISP)

### 3.3.1 Overview

LISP uses IP-over-IP tunnels deployed between border routers located at different domains. The IP addresses allocated to the external interfaces of the border routers act as Routing Locator (RLOC) addresses for the end systems in the local domain. Since an AS usually groups several border routers, the local Endpoint Identifier (EID) addresses can be reached through multiple RLOC addresses. Hence, LISP separates the overall address space into two parts, where only addresses from the RLOC address space are assigned to the transit Internet. Therefore, only RLOC addresses are routable through the Internet, that is, EID addresses are considered routable only within their local domain. In addition, a number of scaling benefits would be realized by separating the current IP address into two different spaces; among them are:

- Reduction of the routing table size in the Default Free Zone (DFZ).
- More cost-effective multihoming for sites that connect to different service providers.
- Easy renumbering when clients change providers.
- Traffic engineering capabilities.
- Mobility without address changing.

The basic idea is that an EID represents an end-host IP address, while RLOCs represent the IP addresses where end hosts are located. At border routers EIDs are mapped into RLOCs, following a map-and-encap scheme, a basic mechanism of a LISP architecture. The scaling benefits arise when EID addresses are not routable through the Internet — only RLOC addresses are globally routable, allowing efficient aggregation of the RLOC address space. Recent studies show that LISP offers some key advantages.

Data plane performance is described on the example shown in Figure 3.2. When the local end host S with EID address 190.1.1.1 wants to communicate with end host D with EID address 200.1.1.2 in a different domain, the following sequence of events occur in LISP:

1. The first step is the usual lookup of the destination address ED in the DNS.
2. Once ED is obtained, the packets sourced from ESource traverse the domain and reach one of the local border routers. In LISP the latter are referred to as Ingress Tunnel Routers (ITRs).
3. Since only RLOC addresses are globally routable, when an ITR receives packets toward ED, it queries the control plane to retrieve the EDestination-to-RLOC mapping.
4. After the ED-to-RLOC mapping resolution, the ITR encapsulates and tunnels packets between the local RLOC address (ITR address 3.3.3.2 in the example) and the RLOC address retrieved from the mapping system, the Egress Tunnel Router (ETR) address in LISP terminology (either 4.4.4.2 or 10.0.0.2 to ED depending on the mapping).
5. At the destination domain, the ETR decapsulates the packets received through the tunnel and forwards them to ED — which, as mentioned above, is locally routable within the domain. From the first packet received, the ETR caches a new entry, solving in this way the reverse mapping for the packets to be tunneled back from $E_{Destination}$ to $E_{Source}$.

Despite the benefits of deploying the LISP architecture, the proposals for the LISP control plane present major challenges. These challenges lie in the fact that since EIDs are not globally routable through the Internet, a mapping system is necessary between EIDs and RLOCs. LISP does not specify a mandatory mapping system, and as a consequence, different proposals can be found in the recent literature, such as ALT [15], NERD [16] or Map Server [17]. Besides, in [18] we introduced a new control plane for LISP; the new control plane presents an improvement on three aspects respect to the

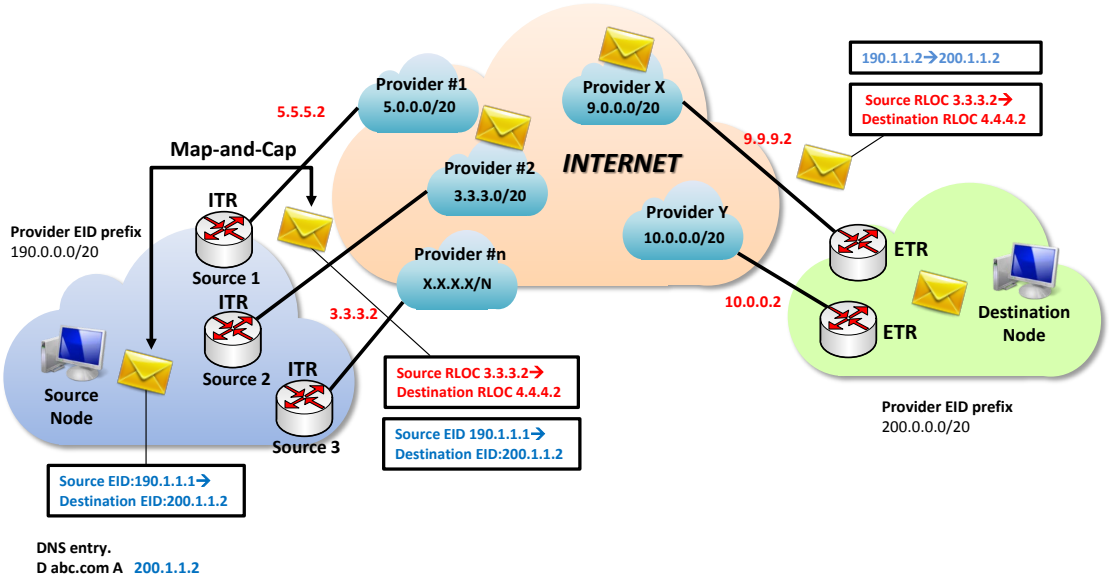


FIGURE 3.2: LISP Fundamentals - Example [2].

existing solutions; (i) firstly, "First packets drop problem" when an ITR does not have a mapping for an EID-prefix; (ii) secondly, potential increase in the latency to start a communication due to the mapping resolution; and (iii) in order to avoid a two-way mapping resolution, the ITR is used as the local ETR for the packets sent from D to S. The latter introduces limitations in terms of inbound Traffic Engineering, especially, when outbound and inbound traffic policies do not match. Despite these improvements there are other issues relating to reachability and reliability that have not been resolved, such as those motivated by an inter-domain link failure.

# Chapter 4

# Semantics and Information Extraction

*This chapter aims to review the fundamentals of Information Extraction (IE) and most importantly, of the sub-field of Ontology-Based Information Extraction (OBIE) in an effort to unveil the possibilities that this research field offers for automating the configuration of network devices based on Command Line Interfaces (CLIs).*

## 4.1 Traditional Information Extraction (IE)

Information Extraction (IE)—a form of natural language processing—aims to automatically identify relevant types of information (typically, entities and relations) from natural language text. Overall, IE does not attempt a comprehensive analysis of a complete document, but instead, to analyze relevant information—i.e., that of interest to an user or application—and thus, ignore other types of data. The main motivation is to obtain structured and machine-processable facts from an unstructured corpus for further analysis. Information Extraction systems have been extensively applied in many research domains wherein textual resources based on natural language are largely available. Consider for instance an Information Extraction system which takes the Web as source of knowledge to extract economic indicators, determine social trends, etc.

Regarding the types of information that can be extracted, authors in [146] distinguish among, *entities*, *mentions*, *descriptions*, *relations* and *events*. The *entity* recognition task refers to the identification of information units in the text (e.g., names, organizations, expressions of time, etc.). *Mentions* are direct or indirect references to the same entities—this task typically involves coreference resolution. *Description* extraction task

enables the identification of descriptive information of an entity, *relations* unveil semantic associations between entities and finally, the *event* identification task—one of the most complex IE task—refers to events involving entities.

Furthermore, the methods for Information Extraction range from rule-based (i.e., manual patterns) to statistical machine learning approaches. The former assumes that models for the IE process are manually formalized by a domain expert, while the later relies on supervised learning algorithms (e.g., support vector machine, maximum entropy, Hidden Markov models, conditional random fields, etc.). According to discussions led by authors in [147] though statistical methods are the most recent trend in IE, both pattern-based and statistical models co-exist and its suitability actually depends on the application and domain-specifics.

## 4.2 Ontologies and Information Extraction

With the advent of the Semantic Web, *ontologies* have emerged as a new paradigm to formalize the knowledge and meaning of a given domain of interest. According to [148], an ontology is defined as *"a formal and explicit specification of a shared conceptualization"*. Overall, the use of ontologies enable *(i)* shared and common reference models of the domain's knowledge, *(ii)* reuse of information across application domains, *(iii)* inference and context reasoning over information, and *(iv)* data disambiguation.

In the field of Artificial Intelligence, most recent knowledge systems rely on ontologies to support the task of Information Extraction—this subfield of IE is best known as Ontology-Based Information Extraction (OBIE). The general notion of an OBIE System is that an ontology provides a predefined model of the information to be extracted. Accordingly, the OBIE system has the ability to link natural textual resources to formal semantic models. Authors in [3] define an OBIE System as "*a system that processes unstructured or semi-structured natural language text through a mechanism guided by ontologies to extract certain types of information and presents the output using ontologies.*" Overall, the design of an OBIE system poses two main challenges, *(i)* identifying entities from the ontology in natural language text and *(ii)* populating the domain ontology, which refers to extracting instances and property values from the text with respect to classes and properties of the domain ontology. According to the study led by authors in [3], the main characteristics of an OBIE System—and which consistently differentiate it from traditional IE—can be summarized as follows:

- *The system's input is restricted to unstructured or semi-structured* ***natural language*** *text.* Those systems using multimedia content (e.g., any combination of

text, audio, images or videos) as source of knowledge are not considered OBIE Systems.

- *The target output is an ontology.* However, it is not strictly required an ontology as the input to the system. In some cases, the OBIE System builds the ontology to be used during the information extraction process itself. Furthermore, OBIE systems most likely rely on domain ontologies, since IE is mainly concerned with the task of identifying concepts from an specific domain. However, if the nature of the developed system is domain-independent it provides seamless support over a wide range of domains.

- *Use of an Information Extraction process* ***guided*** *by an ontology.* The IE process is *guided* by an ontology in order to extract classes, properties and instances, meaning that, traditional methods for IE can be adapted to identify components of an ontology.

Figure 4.1 depicts a general (i.e., common) architecture for an OBIE system. As it is expected variations of this architecture can come with specific implementations, as such, not all components will be necessarily present. For instance, the *Ontology Generator* component of the architecture only exists for those systems in which the ontology is built through the IE process.

Although OBIE is a relatively new field of research, its potential lies in the possibilities of automating the process of information extraction from large corpora. The advantage of such approach is that with the ever-increasing size of corporations documentation or Web-Based information sources, manual processing becomes increasingly complex. Moreover, OBIE boosts the creation of content for the Semantic Web, i.e., automate the generation of meta-data for the Web and finally, it can potentially provide the means to assess the quality of an ontology and further improve it. In the context of this thesis, we propose the use of OBIE from the Command Line Interface as a natural language-based source of knowledge for the configuration of network devices.

## 4.3 Semantic Measure of Relatedness and Similarity

The need of determining semantic *similarity* or *relatedness* between concepts based on ontological structures is becoming a task of utter importance in the field of IE as a means to unambiguously interpret text. Many applications dealing with Natural Language Processing and knowledge management have proven to significantly benefit from the estimation of semantic likeness between concepts. Consider for instance, applications

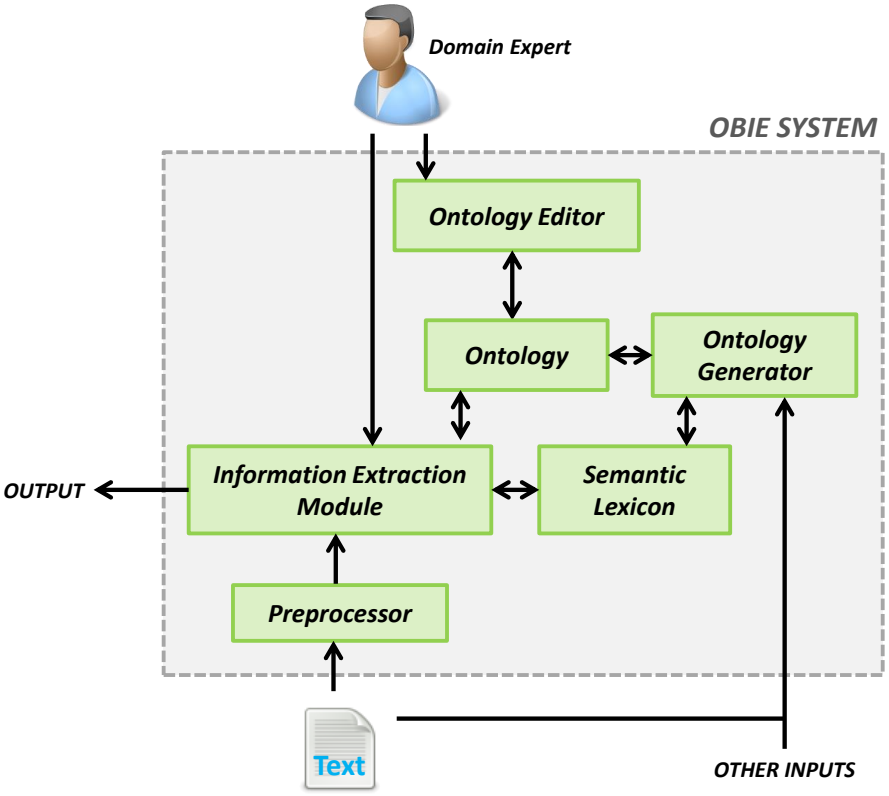


FIGURE 4.1: General Architecture of an OBIE System [3].

based on semantic measures for the purpose of word sense disambiguation [149], synonym detection [150], thesauri generation [151], information extraction and retrieval [152] [153] [154], redundancy detection [155], etc.

Notice that, semantic *similarity* and *relatedness* are slightly different paradigms. Whereas, *semantic similarity* reflects the closeness in meaning between two concepts—generally based on taxonomic relations (e.g., SNMP and OSPF are similar to the extend that they are both networking protocols); *semantic relatedness* reveals the strength of association between two concepts—generally considering both taxonomic and non-taxonomic relations. Overall, semantic similarity is a particular case of relatedness. Two semantically related concepts are not necessarily similar, e.g., consider the concepts *router* and *network* or *interface* and *ip address*—which though strongly related are completely dissimilar concepts.

Semantic measures—both, relatedness and similarity—are classified according to the nature of the computing methods into, *(i)* path-based, *(ii)* Information Content-based, *(iii)* gloss-based and *(iv)* vector-based [156]. The rationale of path or edge-based methods is that relatedness/similarity is computed based on the distance between ontological

nodes. The general notion is that the longer the path between concepts, the more semantically far terms are. The majority of approaches in this category are restricted to taxonomic relations, therefore, they are more of a measure of similarity rather than relatedness. Moreover, Information-Content based methods consider information distribution of concepts, i.e., the amount of shared information. These methods generally introduce corpus statistics. Gloss-based approaches consider as metric the overlap of words between glosses of a concept and finally, vector-based approaches rely on feature vectors for data representation rather than using co-ocurrence counts or contexts. For a detailed survey on semantic measures and their categorizations we refer readers to [156].

Overall, the aim of these measures is to numerically score the semantic proximity of concepts. Most of the developed methods are domain independent, meaning that, they can be adapted in multiple domains. However, given the nature of the target problem, one can significantly benefit from a domain-dependent approach for semantic relatedness/similarity computation. In the literature, many semantic measures have been proposed—with special focus on those targeting semantic similarity. However, robust computation of ontology-based semantic relatedness has just recently gained attention—still remaining a challenging task for the research community [157] [156].

# Part III

# SEMANTIC-BASED CONFIGURATION MANAGEMENT

# Chapter 5

# Ontology-Based Information Extraction System from the CLI

*As briefly introduced in Chapter 1.1.1—in spite of the numerous efforts toward achieving seamless configuration of network devices in multi-vendor (heterogeneous) environments—Command-Line Interfaces (CLIs) continue to be—as far—the preferred mechanisms of network administrators. Unquestionably, the extended use of proprietary interfaces hinder the automation of network device configuration, and furthermore, has serious implications in the field of autonomic network management.*

*Based on the conviction that developing new ways of managing the network or adding new protocols that boost complexity is not the right path. We propose to adapt CLIs as considered to be widely known mechanisms which have set the flagship in this regard. In this chapter, we present an Ontology-Based Information Extraction (OBIE) System from the CLI of network devices, which aims to automatically extract configuration knowledge in order to enable semantic interoperability among heterogeneous configuration environments. We first describe the rationale supporting our approach and then introduce the general architecture of our system. We provide insights on the system's functional modes and modules and finally, describe in detail the specified domain ontology.*

## 5.1 The Rationale supporting the OBIE System

One of the most relevant aspects of CLIs is that—because of its human (manual) oriented nature—it is largely based on textual resources. CLIs are often large and heterogeneous in content, structure and semantics. However, they provide explicit information—encoded in the form of natural language—aimed to help and guide network administrators in the manual use and interpretation of the configuration environment. Moreover, the hierarchical arrangement of the CLI carries implicit knowledge on the semantic relations between commands. Typically, commands in upper levels create the context for following subcommands, i.e., subsequent levels in the hierarchy are semantically related either because they become specifications or direct properties (attributes) of upper levels. Indeed—with the appropriate tools—this information, both explicit and implicit, could be automatically acquired and transformed into useful configuration knowledge.

The notion of automatically extracting knowledge and semantics from already available textual resources has been a field of increased interest among researchers of many other domains—wherein digital information is largely available. Consider for instance, the extraction of information from the Web [158, 159, 160], newspapers [161], twitter [162], résumés [163], medical records [164], etc., to enable numerous applications, such as, question-answering, decision support systems, word sense disambiguation, etc. In light of this, Information Extraction (IE)—a form of natural language analysis—positions as a promising research path to automatically find and retrieve relevant information from the CLI for configuration purposes. Furthermore, Ontology-Based Information Extraction (OBIE) [3]—a recently emerged sub-field of IE—which incorporates the use of a formal ontology can significantly help to improve the IE process. To the best of our knowledge, CLIs have never been explored from a semantic nor IE perspective in the network management arena. For more background knowledge on IE and OBIE please refer to Chapter 4.

From a macro perspective, the design of an OBIE system poses two main challenges: ***(i)*** defining a formal knowledge model of the network device configuration domain (i.e., the domain ontology) and ***(ii)*** developing a learning approach for IE from the configuration Command-Line. Regarding the first challenge, we have created our own structured knowledge-base of the switch/router configuration space, which we have named, **O**ntology for **N**etwork **D**evice **C**onfiguration (ONDC) [165] (cf., Section 5.3.2). ONDC formally specifies the most relevant concepts of the domain and integrates a lexicon of the networking vocabulary. In the context of our approach, this ontology provides a comprehensive and vendor-neutral coverage of the device configuration knowledge. Overall, concepts in the ontology conform to networking standards and well-known technical terminology, reflecting the knowledge of the networking domain regardless of vendors'

specifics. The main motivation for this, is that—apart from proprietary technologies—the vast majority of protocols and features to be set on a network device are common across multi-vendor platforms, otherwise, there would be no means to provide network interoperability. In light of this, configuration capabilities are just about the same for all devices, what essentially changes is the way in which vendors express this knowledge in their CLIs. The differences among CLI-based environments are basically determined by, ***(i)*** the granularity of the tree structure, ***(ii)*** the arrangement of commands in the hierarchy, ***(iii)*** the syntax—e.g., the use of different terminologies for expressing the same concepts—and ***(iv)*** the semantics—e.g., the use of the same terminologies for expressing different concepts (i.e., meaning). In light of all this, we require a solution capable of exploiting this information to unambiguously determine the semantics of each space. This leads us to the second challenge, that is, developing an approach for IE which allows to reconcile the differences between heterogeneous CLIs (cf., Chapter 6). Because of the differentiating features of CLIs, we developed a methodology which not only exploits the (explicit) knowledge given in the form of natural textual language, but moreover, we actually exploit the structure of the CLI itself, i.e., implicit knowledge.

## 5.2 Related Work

*Overall, in this section we will review related work in the field of network device configuration management. In Section 5.2.1, we will focus on the efforts led by industry and key standardization bodies toward seamless device configuration and briefly discuss on the reasons why initiatives such as NETCONF or OF-CONFIG have failed to fulfill the configuration needs of network administrators. In Section 5.2.2 we will discuss on semantic-based approaches toward configuration management. To this end, we first review approaches for device configuration in non-related networking domains and then those in the scope of networking. We clearly distinguish among efforts for network configuration as a whole and underlying configuration at the network element level.*

### 5.2.1 The alternative path of Industry and Standardization Bodies toward seamless network device configuration

As already mentioned, the seamless configuration of network devices is yet far from being accomplished in the networking domain. Based on the limitations of CLI-based mechanisms for the configuration of network devices—in the context of heterogeneous networks—telecom providers have raised their concerns regarding the need of a practical and scalable solution to this largely unsolved issue. The fact that this problem is

still in force is mainly because, in the absence of standard mechanisms, vendor's best interests have imposed. Indeed, proprietary approaches are an effective way of distinguishing from others and keep the lead in a such competitive market. Nevertheless, in the recent years operators' interest for enabling interoperability in such contexts has pushed toward different lines of work. These efforts have emerged both from industry and standardization bodies.

The Network Configuration Protocol (NETCONF) [10] is the most recent effort of the Internet Engineering Task Force (IETF) to develop a *standard* network configuration protocol. NETCONF aims to fill the gap in network device configuration by featuring state maintenance, concurrency, configuration locking, transactional-safe operations across multiple devices, automatic roll-back and operation ordering, distinction between configuration and operational data and consistency in a standard and practical way of use. The major limitation to the adoption of NETCONF is the lack of comprehensive and widely accepted data models, i.e., absence of standards for network management information definition (e.g., to formally specify interfaces, access-lists, ports, etc.). The lack of a data modeling language pushes the interoperability problem to a new level. Initially, several data models emerged as potential candidates for network management definition [51] [50] [52], but neither of them found a path in practice, either because of their increased complexity or lack of semantics. Other NETCONF implementations were built over proprietary data models which in turn raised clear interoperability issues. Most recently, YANG [13]—an IETF proposed standard—has become the most solid contribution for a common data model language for NETCONF. However, almost a decade later and in spite of initial implementations of NETCONF and YANG, only few YANG data models have found broad acceptance [14] and CLIs continue to be the preferred protocol for network device configuration among network administrators [12]. It is clear that there is still a long path before YANG and NETCONF truly become leading standards in the network configuration arena.

Moreover, with network programmability in the spotlight, Software-Defined Networks (SDNs) have rapidly become a major trend in the ICT field. Telecom providers and device vendors concur that SDNs will lay down the foundation for next-generation networks, given their potential for achieving higher flexibility and openness while dramatically reducing costs. A paradigm shift of this nature can clearly transform network management practices, and pave the way for reaching the desired goal of network automation. The OpenFlow protocol [112], and most recently the OpenFlow Management and Configuration protocol (OF-CONFIG) [166], have become key components for controlling and managing SDNs. OpenFlow standardizes the interactions between an SDN controller and the switches under its control. However, it does not provide the functions

that are required for configuring queues, ports, assigning IP addresses or any other configuration toward the device. The OF-CONFIG protocol was recently defined by the Open Networking Foundation (ONF) precisely to that end. A crucial part of the OF-CONFIG specification is that the configurations are transported on NETCONF [10]—which as stated previously also provides mechanisms for the configuration of devices in traditional networks. Unfortunately, as NETCONF itself has not gained momentum yet, it remains to be seen if it will finally become the protocol of choice [11].

In [167] authors discuss on the potential benefits of SDN and OpenFlow to improve various aspects of network management. They suggest that these technologies cannot only ease configuration through software programmability instead of fixed set of commands but also benefit from centralized management. Undoubtedly, OpenFlow and SDNs are key to improve and ease overall network management [167]. However, its flow-oriented nature makes it unsuitable to resolve basic network management operations. This is the case for configurations targeting the router per se (e.g., setting user access rights, such as a user password, etc.) and not actually things going "through" the router—for which it would be indeed largely effective.

Another relevant aspect is that SDN is certainly not necessary for all parts of the network [168]. Moreover, a full replacement of the underlying infrastructure is neither affordable nor feasible for many administrators, which indicates that SDNs will need to coexist and interact with traditional networks for several years. For this reason, hybrid approaches to SDN—a mix of SDN-enabled and traditional network devices—are positioned as strong candidates to ease the transition to new and more flexible network environments [169, 170]. Accordingly, legacy infrastructures will continue to play a crucial role in the SDN future, and will likely give place to new challenges and opportunities in the management field. Although OpenFlow and the elementary configurations supported in OF-CONFIG can suffice for managing OpenFlow devices, network configuration tasks clearly entail much more than configuring flows. Consider for instance requirements such as the configuration of user-names and administrative privileges for authenticated access through CLIs, the configuration of a link-state routing protocol (*e.g.*, OSPF), or a switching protocol (*e.g.* MPLS). Accordingly, the heterogeneity of hybrid SDNs will require of a flexible management model where configurations are not only performed on a per-flow basis.

In a different line of work—aside from standards—industry has approached the device configuration issue by developing dedicated software agents as a means to force static mappings between commands of different configuration spaces. The downside of such approach is that it is not scalable in the context of dynamic environments and requires skillful development teams dedicated to update and maintain these static agents.

Moreover, to overcome the complexities in network management [9] the research community has also explored the field of semantic technologies from Artificial Intelligence (AI) as a promising path to achieve interoperability in domains where standards have not succeeded. Next, we will review semantic-based research approaches in configuration management.

### 5.2.2 A semantic-based path toward seamless configuration management

***Device Configuration in non-related domains***

Configuration management is not restricted to networking devices and is in fact an essential task to many other domains, for example, the case of configuration in the scope of smart homes, manufacturing, e-commerce or service deployment. There is clear evidence on the use of ontologies and other semantic technologies to address the configuration issue in these domains [171] [172] [173] [174] [175] [176] [177]. It is clear that, due to the inherent differences and dissimilar requirements between domains the configuration issue is approached from different perspectives. Nevertheless, the overall aim is to automate the configuration process by using ontologies to formally represent the domain's knowledge, as a means to support reasoning and enable reuse of shared conceptualizations.

The research work led by authors in [171] [172] introduces a novel holistic system for intelligent Smart Home (SH) environments to support device auto-configuration and intelligent control under energy efficiency requirements. Their solution is based on an ontology framework, capable of providing efficient control logics and intelligent decision making. In addition, they develop a semantic extension to the standard Universal Plug and Play (UPnP) protocol to enable communication capabilities of intelligent devices in the context of SHs.

In [173] authors develop a recommender system to support requirement elicitation in a product configuration system. They capture customer requirements represented in an OWL-ontology and assess consistency with respect to manufacturers' specifications (constraints) represented in the Semantic Web Rule Language (SWRL). It also identifies mandatory requirements yet not specified by the customer and suggest them as a means to complete the configuration of the product. In a similar line of research, the work presented in [175] [176] [177] targets product configuration systems for e-commerce through ontology-based approaches.

Moreover, in [174] authors introduce an ontology-based approach for a product configuration system in the e-business field. The aim is to identify customer requirements—expressed in natural language—and output the configuration design of the product that best meets his needs. Herein, three ontologies were developed to represent, ***(i)*** customer needs, ***(ii)*** product functionalities and ***(iii)*** product configuration. Finally, an ontology mapping approach and the use of a Bayesian Network enable automatic conversion between customer needs and product configuration.

Indeed, the scope of configuration as targeted in the aforementioned research initiatives differs from the low-level aspect of the configuration issue in network device configuration. However, they unveil the potential of semantic technologies to reconcile the differences in configuration environments and support translation from custom requirements (in our case "custom" CLIs) to common shared foundations of the domain knowledge.

### *Network Device Configuration*

In the networking domain, several initiatives have also explored the AI path to target network management [178, 179, 180, 181, 182, 118, 183, 117]. Given the broad scope of network management and the numerous functions that it entails, many of these solutions approach different aspects of the problem; for example, the need to unify underlying network management data models [182], autonomic network management [117, 118], integration of management data [180] or the issue of multi-vendor configuration management [178, 179]. Next, we will survey the state-of-the-art in ontology-based network configuration management solutions. To this end, we classify semantic approaches into two groups, ***(i)*** ontology-based approaches to network configuration and ***(ii)*** ontology-based approaches to network device configuration. The former refers to solutions wherein configuration is devised for the network as a whole, while the later refers to semantic approaches targeting the device configuration issue.

#### *Ontology-based Network Configuration Approaches.*

The approaches grouped within this category abstract from the heterogeneity of device configurations and deal instead with the autonomic configuration of the network—at a higher level [184, 118, 117]. Overall, the aim of these approaches is to provide the network with self-management and context-awareness configuration capabilities. They propose—in different contexts—the use of semantic technologies to provide a smart environment in which configuration services are triggered whenever a network condition is given. To this end, they combine the use of the Ontology Web Language (OWL) and the Semantic Web Rule Language (SWRL) to model the management information and network behavior,

respectively. SWRL enables rule integration into the ontology, so whenever a condition is fulfilled a service is automatically invoked and then executed. In the context of IP networks this service can be represented by a configuration script, which includes the CLI device-specific commands to the requested service. As can be seen, these works are restricted to static device configurations as underlying issues remain unsolved. This means that to ensure scalability and flexibility of their solutions, an approach to resolve device configuration heterogeneity is required.

From academia, several works have emerged in an effort to integrate ontologies and other semantic technologies to achieve interoperability at the semantic level. Next, we will review the status of this research field.

*Ontology-based Network Device Configuration Approaches.*

Within this category we classify ontology-based approaches to the semantic interoperability problem in network device configuration management. In the work presented in [178], authors propose the use of ontologies to describe a NETCONF workflow. However, with the so-mentioned limitations of NETCONF this solution in neither scalable nor practical. Moreover, in the work presented in [179] authors introduce an ontology-driven approach to the semantic interoperability problem in network management and validate it for the case of multi-vendor router configuration. Their major contribution is a generic similarity-based ontology mapping strategy which can be seamlessly applied across the ITU-T Telecommunication Management Network (TMN) layer model. For validating the configuration use case they built vendor-specific ontologies (one per network vendor), wherein CLI commands where properly modeled and classified. They further applied the mapping strategy to a set of selected commands to assess the semantic match between both configuration spaces. Beyond the limitations of the similarity function and computational methods—pointed out by the authors in [179]—scalability of this approach is an issue, inasmuch as, ontologies for CLI environments are not given by vendors' in advance. Thus, formal representations of the CLI knowledge must be manually built by domain experts—a task which is per se sufficiently complex and challenging—and continuously updated as new features, vendors or releases emerge. As if that was not enough, the ontology expert would require to gain expertise in the new CLI environment beforehand. The ideal scenario would be to assume that vendors handle ontologies in advance for every CLI, in a similar way as drivers are provided for every device. Nevertheless, this is a demanding requirement which is far from vendors' roadmap. To the best of our knowledge there are no further efforts in this line. Nevertheless, we firmly believe that ideas from the ontology research arena can still be brought to the

CLI configuration domain to achieve interoperability at the semantic level. The semantic interoperability problem in network configuration management requires a solution capable of adapting to the dynamics of current configuration environments in an easy and automated way.

Despite of the numerous initiatives regarding the application of ontologies and AI-based techniques to approach network management, the exploitation of semantic technologies continues to be a research challenge. There are still many paths to be explored in order to benefit from such technologies. Moreover, none of these works have explored the field of OBIE as a means to enable interoperability in the network configuration domain.

## 5.3 The OBIE System Architecture

Herein, we present the general architecture of an Ontology-Based Information Extraction (OBIE) System from the configuration command-line of network devices. Figure 5.1 depicts the general architecture of our system. Observe that it takes two inputs, namely, ***(i)*** the Command Line Interface (CLI)—as natively provided by vendors (i.e., unprocessed)—and ***(ii)*** the domain ontology (i.e., ONDC)—which has been specified by us and formally defines the knowledge of the network device configuration domain. Furthermore, the output of our system is a device-specific version of the target domain ontology, i.e., the ontology populated with instances of the configuration commands. These device-specific ontologies are further stored in a repository to enable potential functionalities to third-party applications, *e.g.*, a Network Management System (NMS) requesting the commands for setting an interface IP address for dissimilar device models.

Our system's architecture is based on a modular design which accommodates several components into two functional blocks, namely, the *offline* (lower block) and the *online* (upper block) modes (cf., Fig. 5.1). The *offline* functionality is responsible for performing the semantic abstraction of multi-vendor (i.e., heterogeneous) configuration environments. Accordingly, the system's *intelligence* lies within this mode. The **Semantic Learning Engine** (cf., Fig. 5.1)—which represents the core module of this functionality—carries the logic and algorithms for extracting and interpreting the information of the CLIs. In Chapter 6 we will show how the semantic instantiation of commands is done by thoroughly describing our approach for IE from the CLI. Moreover, the *online* functionality provides a web-based interface through which users or third-party systems can retrieve semantic-based configurations for heterogeneous (i.e., dissimilar) network devices. In other words, this mode enables access to the semantic models generated in the offline mode, so external systems or applications can benefit from multi-vendor configuration abstraction. To illustrate the online functionality,

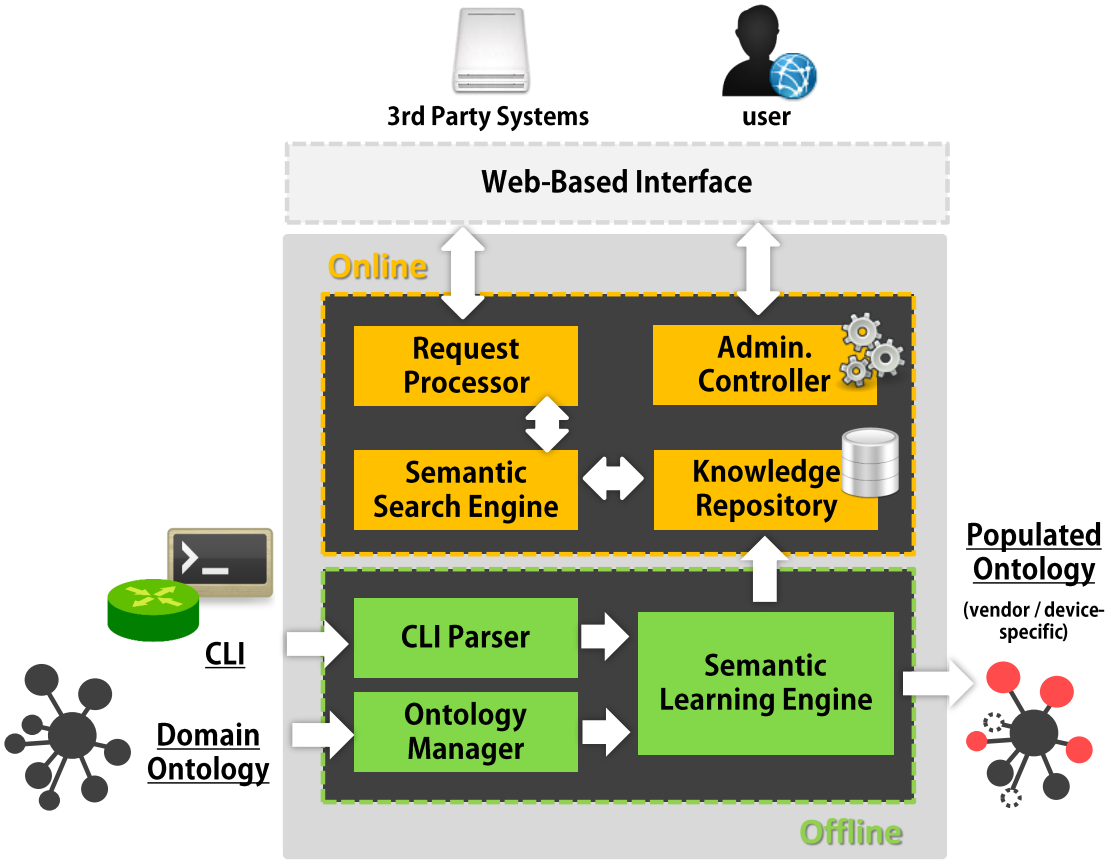


FIGURE 5.1: General Architecture of the OBIE System for the device configuration domain.

consider a system or administrator that must perform the real deployment of an initial network planning strategy, which involves a high number of configurations across multiple devices from different vendors. A process of this nature, could make single requests to our system for each required configuration operation and this would retrieve the commands for each available device model. For instance, request the commands for setting the domain's name for all available devices in the network. Overall, the *offline* and *online* modes aim to mitigate the efforts related to the configuration of devices in heterogeneous environments. The execution of both functionalities completes the configuration cycle, from the semantic definition of the configuration space to the automatic retrieval of commands, regardless of the underlying heterogeneity.

### 5.3.1 System Modules

The basic architectural components of our system are, namely, **(1)** CLI Parser, **(2)** Ontology Manager, **(3)** Semantic Learning Engine, **(4)** Knowledge Repository, **(5)** Administration Controller, **(6)** Request Processor, and **(7)** Semantic Search Engine (cf.

Fig. 5.1). Next, we will provide a detailed description of the main functionalities of each module.

### *CLI Parser*

The **CLI Parser** provides the functions to breakdown the CLI into its structural parts, namely, *commands* (cmd) or *variables* (var) and *help descriptors* (hd) (cf., Fig. 5.2). Though all three structural elements are key to derive the semantics from the CLI, it is of utter importance to distinguish between each type, as only commands and variables are target of instantiation. The information provided in the help descriptors will serve to contextualize, disambiguate and identify relevant concepts—which will ultimately assist in the process of semantic instantiation. The fact that "helps" are aimed to guide network administrators on the manual use and interpretation of the CLI makes this information particularly meaningful to our system, as most likely it provides reference to common (standard) networking terminologies—instead of custom and vendor-related ones. All in all, the semantics provided by vendors must converge at some point to standard terminologies, otherwise, their solutions would be unattainable and impractical, and the learning curve expensive. In this sense, helps are with higher chance points of semantic convergence while commands are more likely subject to vendor customization. Although it is an irrefutable truth that networking-related terms can occasionally be misleading, even if helps do not explicitly provide a reference for disambiguation, the context—*i.e.*, information in contiguous levels—can help determine a term's sense. This feature will be further exploited in our learning approach described later in Chapter 6. The output of this stage is an XML Document which represents the CLI with distinction of its structural components. Figure 5.3 shows a simplified example of the generated XML Document. The XML document consists of a root element "CommandSet". This element contains two different child elements, namely, "Variable" and "Command". The "Variable" and "Command" elements contain the attributes, "exec", which indicates if

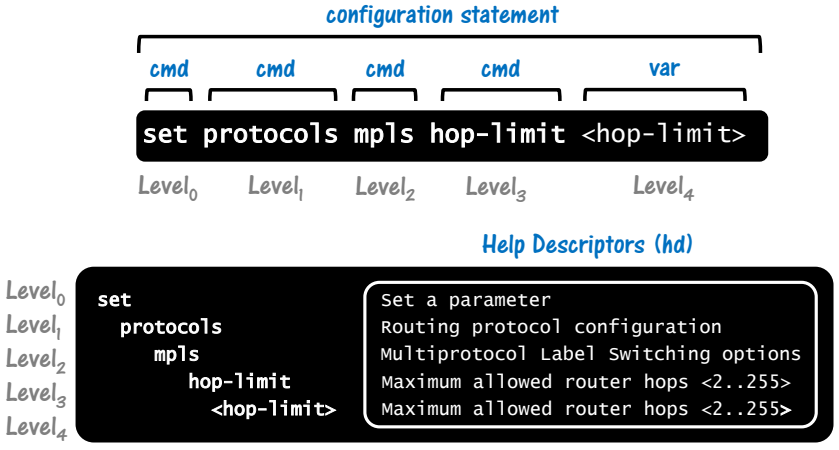


FIGURE 5.2: Typical CLI structure.

a command is executable or not, “help”, which defines the help descriptor provided in the CLI, “metainfo”, which stores any meta-data appended to the element and “name”, which indicates the keywords used to reference the command or variable in the CLI.

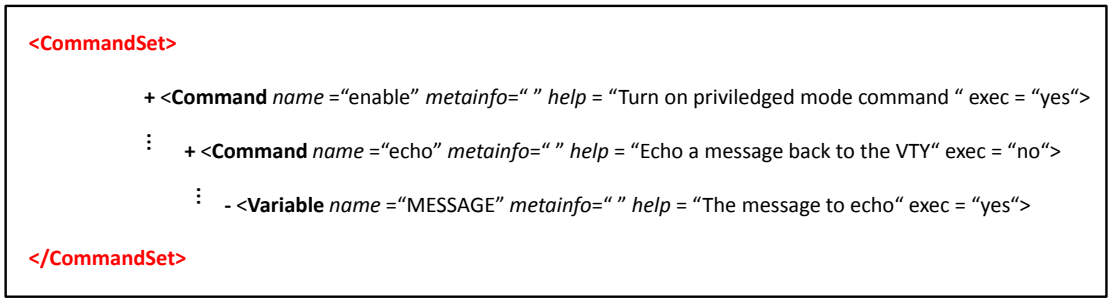


FIGURE 5.3: XML File Example.

Moreover, the CLI Parser also browses through the hierarchy in order to build the sets of executable configuration statements—i.e, valid sequences of commands and variables which semantically represent one or several atomic operations. The reason for this is that given the hierarchical and relational nature of CLIs—in most cases—single commands are not sufficient to provide the complete semantics, instead, it is the combination of commands in the hierarchy which build the meaning of a configuration action. A typical configuration statement is shown in Fig. 5.2.

### *Ontology Manager*

Overall, the **Ontology Manager** provides an interface to the domain ontology and further makes the knowledge available for reasoning and processing to our learning algorithm. Most specifically, it has a two-fold functionality:

- First, to read, access and manipulate OWL constructs—via the OWL API [185]—and create a Java-based graph of the domain ontology which replicates classes, relations, and individuals formally defined within the knowledge model.
- Second, to export all available *atomic* configuration operations derived from the domain ontology via Web Services to third-party applications.

### *Semantic Learning Engine*

The **Semantic Learning Engine** carries the algorithms for the extraction of configuration knowledge from the CLI. This module is the core of our system and comprises the logic which finally makes up to the instantiation of commands into semantic categories.

In other words this module is where the actual information extraction takes place. It can be further differentiated in four components, namely, ***(i)*** Data Pre-processor, ***(ii)*** Lexical Matching, ***(iii)*** Semantic Analysis, and ***(iv)*** Decision Maker (cf., Fig. 5.4). The approach to Information Extraction (IE) will be described in Chapter 6.

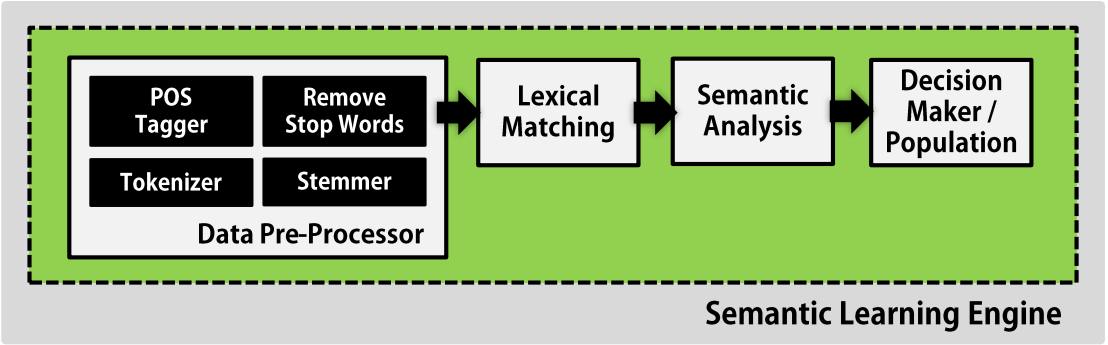


FIGURE 5.4: Semantic Learning Engine Internal Architecture.

***Knowledge Repository***

The **Knowledge Repository** provides storage capabilities for the generated semantic models. Moreover, ontologies are stored on the basis of its heterogeneity, i.e., vendor, model and Operating System (OS) release of the device.

***Administration Controller***

The **Administration Controller** enables user-level administrative control of the system. It provides functions such as, adding, updating or deleting semantic models (i.e., ontologies) from the repository; requesting the system to parse a new CLI or manually manipulating ontologies.

***Request Processor***

The **Request Processor** manages all configuration requests in order to retrieve the corresponding commands for a given semantic operation. The configurations are requested by means of **(a)** atomic operations—which have been exposed by the Ontology Manager—and **(b)** device tuples (vendor, device model and OS version). For instance, consider requesting configuring a DHCP address pool for the Cisco 7200 OS v-12.4 or the HUAWEI NE20E-S.

***Semantic Search Engine***

The **Semantic Search Engine** is the module responsible of building and formatting the valid sequence of commands and variables from the semantic structure.

### 5.3.2 The Domain Ontology for Network Device Configuration (ONDC)

The Ontology for Network Device Configuration (ONDC) acts as a general semantic foundation for the configuration of network devices. As previously stated, it provides a common and shared conceptualization of the configuration knowledge, regardless of vendors' specifics. The main driver for building such an ontology is based on the fact that the knowledge expressed in CLIs—apart from proprietary developments—is mostly related to well-known concepts and technologies. This is mainly because device manufacturers tend to keep their products close to standards, as a means to ease interoperability and comply with regular configuration features. Nevertheless, the use of terminologies to name commands and variables is what most likely differs among vendors —either because different representations are used for the same semantics (a syntactic problem) or the complete opposite, same terminological representations are used for different semantics (a semantic problem). In light of this, it is not what CLIs provide what mostly concerns network administrators, but most importantly, the way in which this knowledge is expressed —*i.e.*, the use of dissimilar terminologies and their corresponding semantics. The ontology was formally defined using the Web Ontology Language (OWL)—the defacto language for encoding knowledge over the Semantic Web—and built with Protègè, a powerful free open-source ontology editing tool and knowledge acquisition system developed by Stanford for the creation, edition and manipulation of ontologies. Moreover, we used the Protègè API to access, create and manipulate ontology resources. The ontology has been defined taking the knowledge provided by networking experts, in addition to information extracted from configuration manuals. For this reason, key concepts and relations of the domain were unambiguously identified and formally encoded. Furthermore, a domain lexicon was integrated in the ontology.

The design of an ontology is closely related to the ultimate use or purpose of the knowledge representation model. In the context of our approach, ONDC constitutes a valuable resource to guide the configuration information extraction from the CLI. In light of this, we have determined the need of defining two distinct layers, namely, the router *resource* layer and the router *operation* layer—very similar to the approach followed by authors in [186]. The notion of a layered structure of the ONDC ontology is depicted in Fig. 5.5. The former defines the entities, concepts and resources of the domain—both, physical (e.g., an interface or a LAN port for the routing domain) and virtual (e.g., a routing

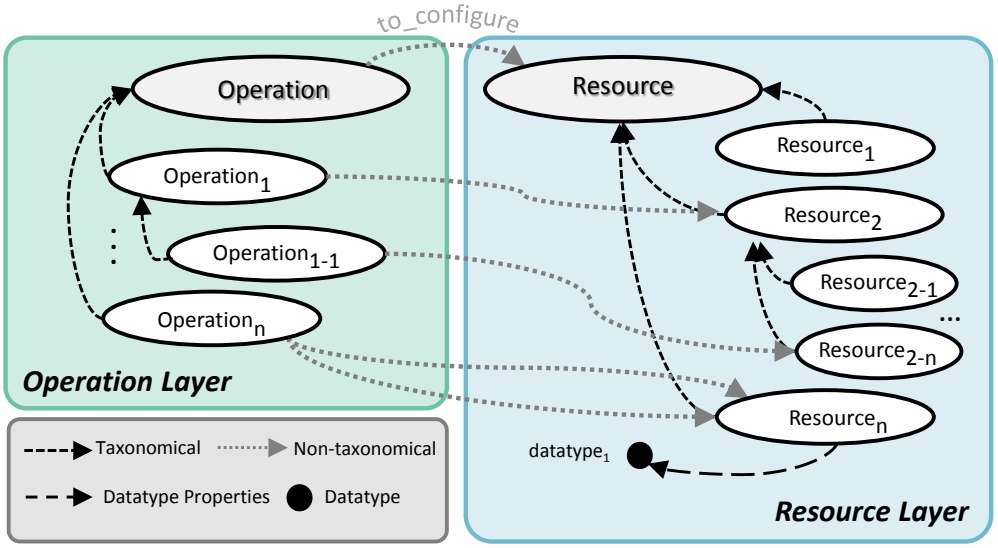


FIGURE 5.5: Layered structure of the domain ontology.

protocol or the OSPF hello interval). Moreover, the latter defines the functional concepts of the domain, i.e., the set of operations that can be performed over virtual and physical resources—e.g., configure a router host-name or remove a static IP route, etc. Notice that concepts in the operation layer are specified in terms of verb phrases (e.g., set, delete, configure, show, etc.) and semantically associated to concepts in the resource layer (cf., Fig. 5.5). In short, a resource represents a component that can be supplied or consumed in an operation.

The resulting ontology represents over 600 resources and near 320 operations [165]. We have developed our ontology, based on the use of all OWL constructs, namely, classes, individuals, properties, restrictions, etc., in order to enrich the domain knowledge model. We have defined hierarchical (*i.e.*, taxonomic *"is-a"* type of relations) and non-hierarchical relationships between concepts, in an effort to improve the information content of the domain. Moreover, we have modeled user-defined data-types using the pattern facet restriction feature of OWL2 to define custom types to match regular expressions. This feature will enable us to validate domain-specific types of data, *e.g.*, to identify ranges or an IPv4 address—which is a 32 bit number expressed by a standard notation of the form 192.45.32.120 where dot separated numbers range from 0 to 255. Notice that the knowledge encoded in the ontology will help us resolve ambiguity. For example, we can determine if the occurrence of a term corresponds to a certain concept by identifying an address format or a measurement unit.

For illustration purposes, from now on we will consider the ontology as a semantic graph $G$ (cf., Fig. 5.6).

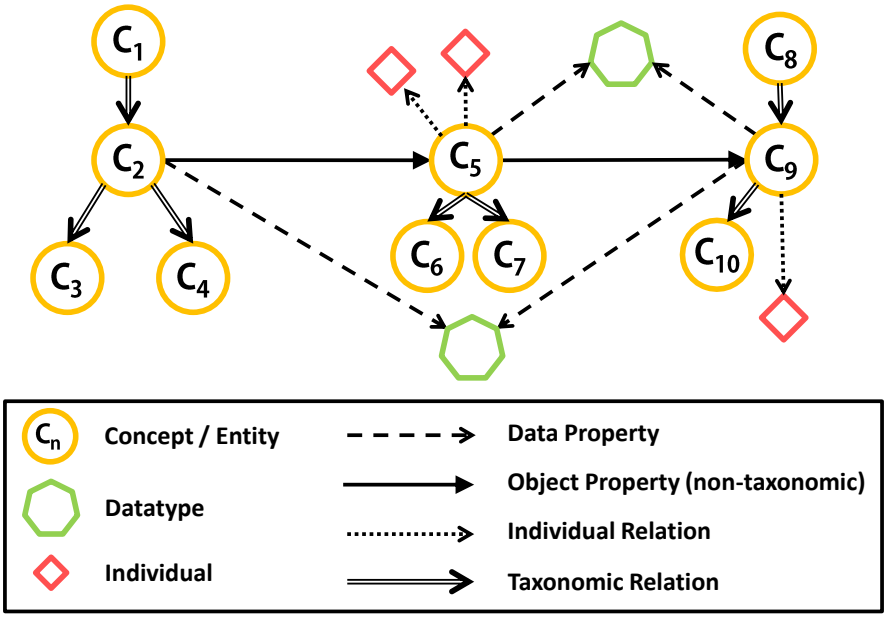


FIGURE 5.6: Ontology modeled as a semantic graph.

*Definition* 1. The **Semantic Network** ($G$) is a directed graph where nodes represent concepts of the networking domain, and edges represent attributes or relations between concepts.

For the sake of readability, we will use the terms "*node*", "*concept*" and "*entity*" interchangeably for the rest of this chapter. Likewise, the terms "*edge*" and "*relation*" will be used interchangeably to refer to the semantic links between ontological concepts. Notice that there are different types of relations, namely, *taxonomic*, which reflect subsumer relations, *non-taxonomic*, which reflect object-type of relations (i.e., between entities), *individual*, which represent membership relations and finally, *data-property*, which represent datatype attributes.

# Chapter 6

# Information Extraction Algorithm

*This chapter aims to thoroughly describe our methodology for configuration knowledge extraction from the CLI. To this end, we first introduce the general notion of the proposed algorithm and then provide a step-by-step description of our two-fold stage methodology.*

## 6.1 The general notion of a two-fold stage algorithm

Figure 6.1 depicts a general diagram of our IE algorithm. To achieve the overall goal of our system we target the process of IE in two stages. In ***Stage I***, we look in the CLI for *(i)* verb phrases and *(ii)* relevant concepts of the switch/router domain (i.e., networking entities) with respect to particular components of the *resource* layer of our domain ontology. In ***Stage II***, we integrate this knowledge (i.e., verbs and domain entities) in an effort to classify commands into their corresponding semantic categories—with respect to the components of the *operation* layer. The methodology we have developed is a multi-step process (cf., Fig. 6.1) which combines the use of NLP techniques and other semantic resources to unveil the semantics of the CLIs. In order to ease explanations, we will consider the configuration statement shown in Fig. 6.2 as example throughout this Chapter. Notice that the referred sequence of commands and variables semantically represents the configuration of an interface's IP address. Next, we will explain each step of our IE algorithm.

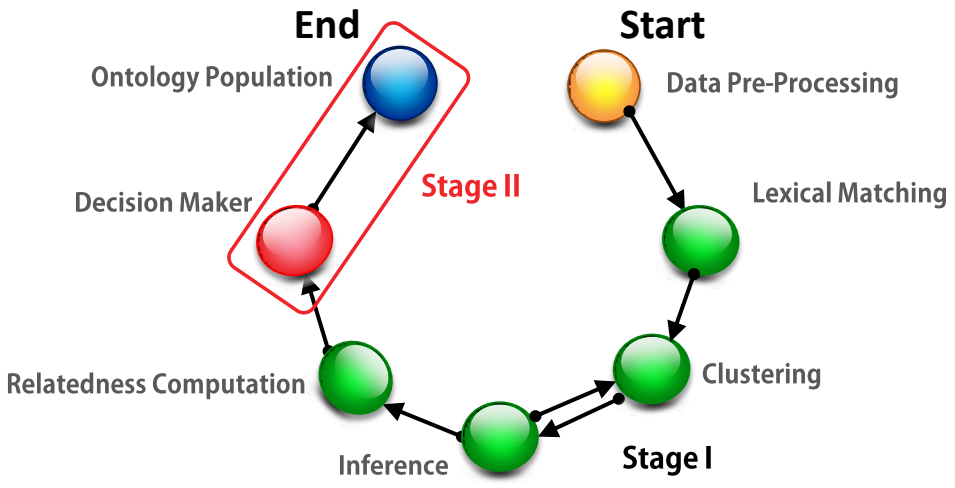


FIGURE 6.1: Configuration Knowledge Extraction Algorithm (Stage I and II).

## 6.2 Stage I: Resource Identification

### 6.2.1 Data Pre-Processor

The first component in the IE workflow is the **Data Pre-Processor** which combines shallow Natural Language Processing (NLP) tools for basic data pre-processing. Overall, Data Pre-processing includes the following resources, namely, ***(i)*** Part-Of-Speech (POS) Tagger, ***(ii)*** Tokenizer and ***(iii)*** Stemmer. The *POS Tagger* resource allows the identification of verb phrases from the CLI. Notice that in-depth POS analysis is far from being required as typical CLIs lack of grammar rules and verbosity. Accordingly, the information present in the CLI is likely to be short and concise—sometimes even insufficient to be self-explanatory. The lack of verbosity and proper grammar restricts the content of CLIs to ***(i)*** concepts (*e.g.*, $\text{Level}_2$ and $\text{Level}_3$ in Fig. 6.2) and ***(ii)*** verb

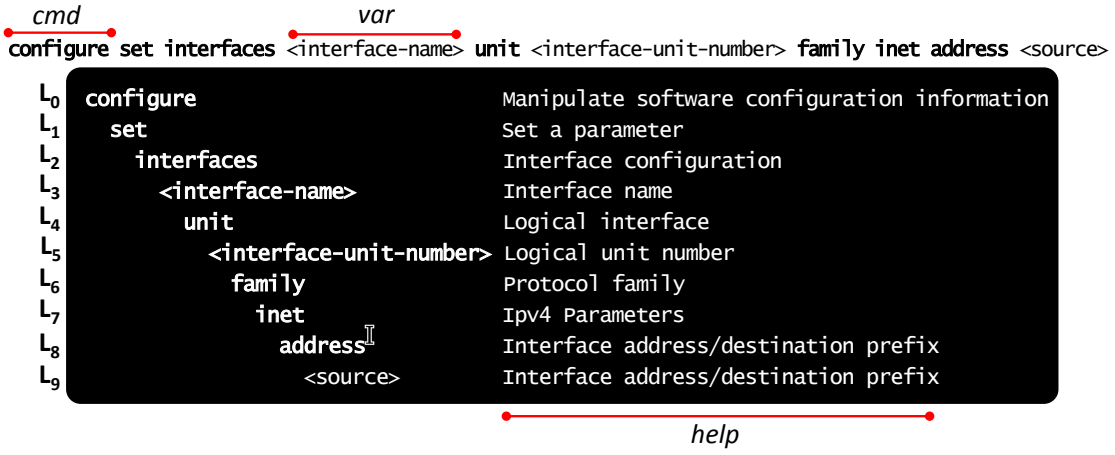


FIGURE 6.2: Example of a router configuration statement.

phrases (*e.g.*, $\text{Level}_1$ in Fig. 6.2). This significantly simplifies the scope of the semantic search. Moreover, the number of configuration actions (i.e., operations) are finite and common across multi-vendor platforms, e.g., set, delete, add, merge, show, load, reset, etc. In our implementation, we have used the Stanford Log-linear POS Tagger [187] for this purpose. The next resource to be applied is the *tokenizer* which separates data into tokens for further processing. We have used the tokenization tool in the Apache OpenNLP library [188]. Notice that commands are typically expressed with single short suggestive keywords as to ease manual configuration. Nevertheless, in the context of CLIs, we can often find the use of hyphenated words to improve the expressiveness of a command—e.g., "source-port", "source-class", etc. In an effort to preserve the semantics of the CLI—in the context of our solution—hyphenated words account as single tokens. Moreover, we remove stop words, i.e., we filter irrelevant words from the CLI data, e.g., articles or prepositions. It is important to highlight the fact that, the number of stop words in our domain is not particularly significant, as commands and variables are single keywords and help descriptors are generally short and concise phrases with poor grammar. However, we have developed a custom list of stop words to avoid returning or processing unnecessary information. The final resource to complete the step of Data Pre-Processing is the *Stemmer*, which allows to reduce inflectional forms of a word to its common base form. We used the Stanford NLP stemmer (CoreNLP) [187]—which is based on the Porter stemming algorithm. Herein—as done by many search engines—words with the same stem are treated as synonyms. This will improve the performance of our system by increasing the probability of successful hits when performing lexical matching.

### 6.2.2 Lexical Matching

The next step in the algorithm is to perform **Lexical Matching**, i.e., we make extractions with respect to particular components of the domain ontology. This strategy fits with the general notion that—overall, and despite CLIs heterogeneity—concepts must converge to well-known technical terminologies, otherwise, it becomes increasingly complex to achieve interoperability and moreover, stay competitive in an industry led by standards. Consider for instance, a device vendor using custom terminologies to refer to standard concepts of the networking domain, e.g., an IP address or a networking protocol (MPLS, DHCP, etc.). In such a context, the interpretation of CLIs becomes overly intricate, unless vendors provide mappings to standard terminologies. In light of all this, the notion of CLIs having to rely on technical (standard) terms—at least those likely to be referents in the field—makes lexical matching a feasible strategy to identify key concepts of the domain. Nevertheless, it is clear that in a field full of terminologies

and ever-changing technologies, there is still space for syntactic and semantic ambiguity, as typically, vendors use different terms to refer to the same concepts or on the contrary, use the same terms to refer to different concepts.

Consider the graph-based representation of the ontology (cf., Fig. 6.3 (a)), the lexical matching stage results in the activation or highlight of nodes and links of the Semantic Graph $G$ for every identified entity in the CLI—the notion of "activated" nodes is depicted in Fig. 6.3 (b)). Notice that we admit both, partial and exact matching. If an exact match is found for a given term, partial matches are discarded. In the case for which a term matches several ontological concepts and these are taxonomically related, we hold the Least Common Subsumer (LCS), *i.e.*, the most concrete taxonomic ancestor. In other words, we generalize in the absence of information. To illustrate this, consider $\text{Level}_2$ in our example. The CLI information for this level is "Interface Configuration". Accordingly, candidate concepts (lexical matching) will be: $\langle interface \rangle$, $\langle ethernet - interface \rangle$, $\langle 100G - Interface \rangle$, $\langle 10G - Interface \rangle$, and $\langle interface - name \rangle$. From the ontological structure, we know that these concepts have a subsumption relationship. The $\langle ethernet - interface \rangle$, $\langle 100G - Interface \rangle$ and $\langle 10G - Interface \rangle$ concepts are subtypes of $\langle interface \rangle$, while $\langle interface - name \rangle$ is an attribute (*i.e.*, an ontological data property) of the latter. In the absence of information we select the LCS, *i.e.*, we generalize to the $\langle interface \rangle$ concept. Moreover, if exclusive properties of a concept are discovered in subsequent levels, we can further select a specification of the concept. This functionality is performed by the inference stage of the Semantic Analysis. Notice that there are cases for which candidate concepts do not have a LCS. In these cases, concepts are considered disjoint, i.e., only one can accurately define the semantics of the given CLI term. Further clustering and semantic relatedness will aid in the disambiguation for these scenarios. To illustrate this case consider the following. For $\text{Level}_8$ in our example, the lack of verbosity in the CLI can generate ambiguity between the ontological concepts, $\langle mac - \underline{address} \rangle$ and $\langle ip - \underline{address} \rangle$. Although in principle we lack information to disambiguate between both concepts from a lexical perspective, we know in advance that only one can properly define the semantics of the given term. For this reason, we identify them as disjoint concepts and further semantic analysis will allow us to select the best candidate concept.

It is important to realize that even if concepts are not identified by lexical matching, mainly because of the use of custom or dissimilar terminologies, while performing **Semantic Analysis** we can identify relevant concepts by inference. Therefore, we do not make limited use of the ontology—such as names of classes—moreover, we use the ontological structure to enhance our assessment. The Semantic Analysis stage can be further differentiated into, clustering, inference and semantic relatedness computation, as shown in Fig. 6.3.

### 6.2.3 Clustering and Inference

In this step activated resources are grouped into semantic clusters (cf., Fig. 6.3(c) and 6.4). We form clusters between *directly* connected resources of adjacent levels (e.g., $\{C_2, C_3\} \in G$). If an activated resource is disconnected to other active concepts in the graph, it uniquely forms a cluster (e.g., $\{C_1, C_4\} \in G$). Notice that the notion of nodes being part of fully interconnected clusters is based on the premise that commands are arranged in the hierarchy by association—i.e., commands become more specific down in the tree structure. Accordingly, the concepts that derive from commands and variables in contiguous levels are expected to be semantically related to a certain extent—directly or not. Ideally, activated resources would form part of a single cluster (i.e., interconnected concepts), however, the degree to which entities are actually related will also depend on the granularity of the CLI—which varies for every vendor. For this reason, clustering is not sufficient and we require other means to measure (i.e., quantify) the degree to which concepts are semantically related, this is part of the *relatedness computation* stage, which will be described later in this section. Notice that we restrict clusters to non-disjoint nodes. This means that concepts that have been triggered by the same activation keywords necessarily belong to different clusters—even if they are directly connected—as they are (from a lexical perspective) equally likely candidates for the

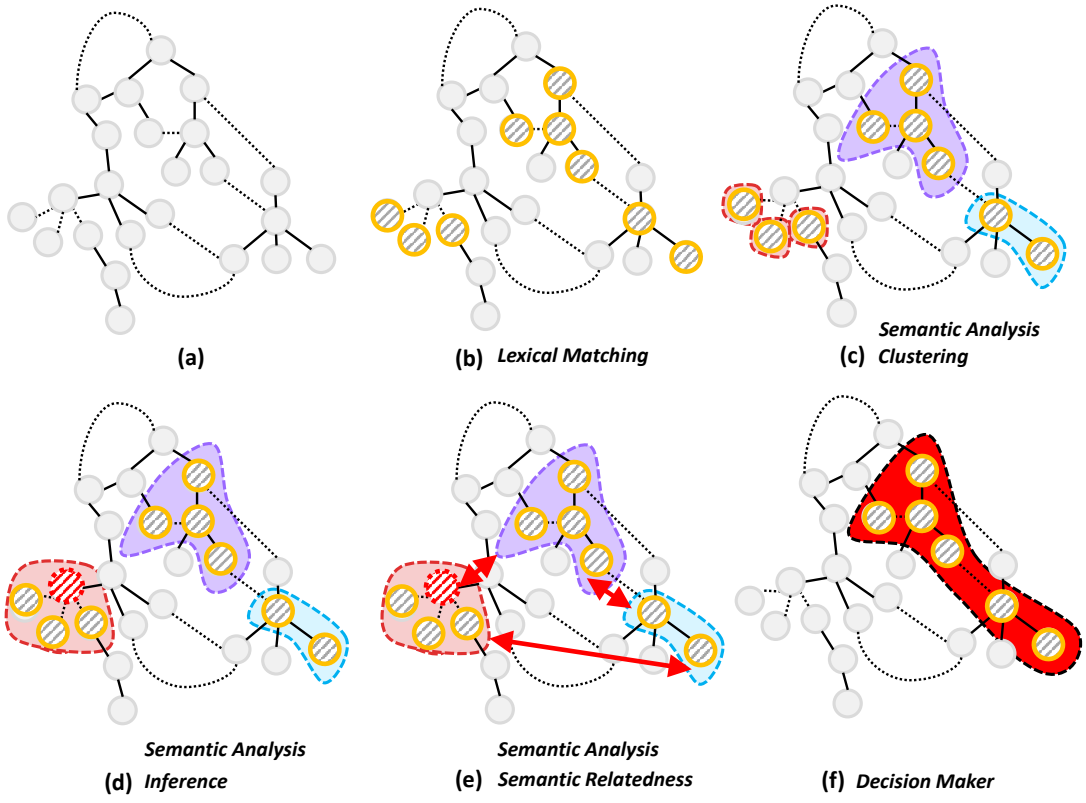


FIGURE 6.3: Semantic Analysis: Step by Step Diagram.

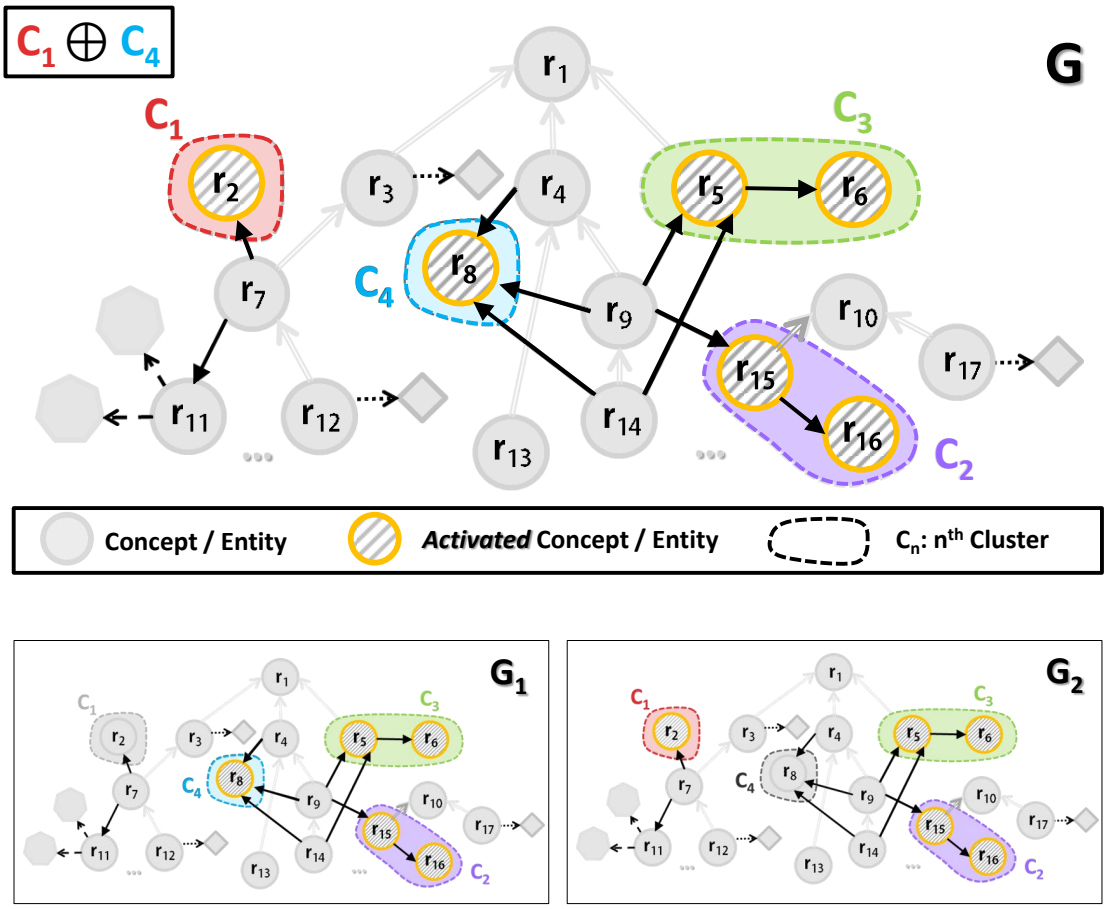


FIGURE 6.4: General Notion of the Clustering Stage.

same set of concepts. For instance, consider in Fig. 6.4 resources $r_2$ and $r_8$ to be candidate concepts for the same set of keywords. Accordingly, clusters $C_1$ and $C_4$ are disjoint, as only one can fairly represent the semantics of the given term(s) and thus, each belong to a different subgraph—i.e., valid combinations of non-disjoint clusters (e.g., $G_1$ and $G_2$). In further stages, semantic relatedness is computed over each subgraph in an effort to promote closest nodes (i.e., those with higher density) as the most suitable concepts for defining a given configuration statement.

Furthermore, we perform *semantic inference* (cf., Fig. 6.5) as a means to derive knowledge that is not explicitly expressed in the CLI. To this end, we exploit the ontological structure and reason over the facts and axioms formally defined in the router/switch configuration domain ontology. The inference stage has a two-fold purpose. First, to discover potential concepts that were not identified in previous stages, either because of *(i)* the use of very dissimilar terminologies—i.e., use of a vocabulary which is not well aligned to the domain lexicon—or *(ii)* because of the granularity of the hierarchy. For instance, for less granular hierarchies, knowledge is most likely to be implicit and thus, concepts can fail to be identified. To illustrate the inference stage, consider the example shown in

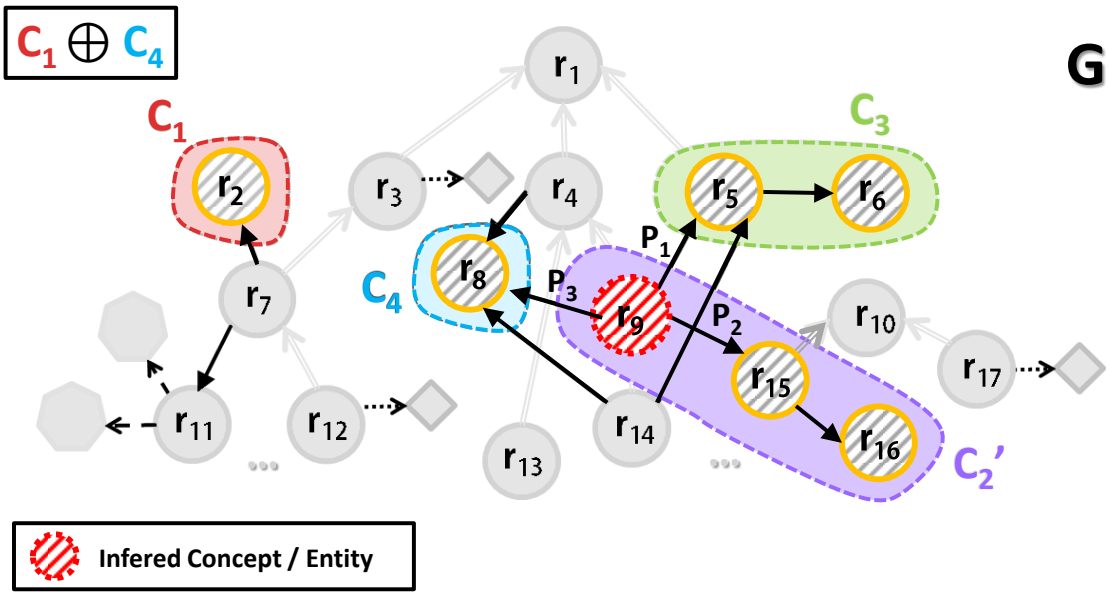


FIGURE 6.5: General Notion of the Inference Stage.

Fig. 6.5. If we identify from the CLI the concepts "$\langle administrative - distance \rangle$" ($r_8$), "$\langle destination - prefix \rangle$" ($r_5$) and "$\langle bandwidth \rangle$" ($r_{15}$), we can infer from the equivalent axioms of the ontology that most likely we are referring to the concept "$\langle route \rangle$" ($r_9$), for which these 3 concepts are exclusive properties. Based on this, we can further activate the inferred resource ($r_9$) and build the semantics of the given configuration statement. The second purpose of this stage is to generalize or specify already active concepts by taking into account contextual information. Consider the following example, if the concept "$\langle routing - protocol \rangle$" has been identified for a given level of the hierarchy and we then identify the concept "$\langle OSPF - area \rangle$", we can infer that we are referring to the OSPF protocol—which is both a routing-protocol and exclusively related to the attribute "$\langle OSPF - area \rangle$". In any of both cases, if this stage results in the activation of a node by inference (e.g., $r_9$ in Fig. 6.5) we perform clustering once more to group nodes by direct association (e.g., $C_2'$ in Fig. 6.5).

### 6.2.4 Semantic Relatedness

The rationale of computing **semantic relatedness** between candidate concepts is to promote closest nodes. This is based on the premise that because CLIs are arranged by association, successive commands in the hierarchy are highly interrelated concepts of the domain. To exemplify this general notion, let us consider the example shown in Fig. 6.6, where the resources *"MAC_Address"* in cluster $\mathcal{C}_C$ and *"IP_Address"* in cluster $\mathcal{C}_B$ are candidates for the CLI term "address". Observe that, both resources—and accordingly, the clusters to which they belong to—are disjoint, as only one ontological

class is expected to accurately represent the semantics of the term. From a lexical perspective, the succinctness of the CLI (i.e., lack of information) is what actually leads to the ambiguity between both concepts. However, based on the contextual background, the concept *"IP_Address"* seems to be a better candidate, as it is semantically related to a higher extent to concepts identified in adjacent levels (*i.e.*, higher node density).

Overall, relatedness will contribute to information extraction and the identification of potential outliers (command sense disambiguation)—i.e., picking the most suitable sense of the word and constraining the interpretation of terms in our system. In the context of our solution, we define ***relatedness*** ($\mathcal{R}$) as a function that computes the strength of semantic association between a set of clusters $\{\mathcal{C}_k\} \in G_k$. Notice that, in contrast to state of the art approaches, $\mathcal{R}$ is not restricted to a given "pair" of concepts, but instead, extends to reflect the proximity in meaning of a "set" of concepts. Typical existing measures of relatedness based on ontologies exploit only taxonomic relations, accordingly, they are more a measure of similarity rather than relatedness. In light of this, our measure $\mathcal{R}$ is computed by interpreting the paths between clusters, based on both taxonomic and non-taxonomic relations. In the next lines, we will explain in detail our relatedness measurement $\mathcal{R}$.

Let $G(\mathcal{C}, R)$ be a directed graph, where the vertex $\mathcal{C}$ represents a cluster $\in G$, and the edge $R$ represents a relationship among two adjacent clusters (cf., Fig. 6.7). Let $G_k \subseteq G$ represent a connected subgraph of $G$, and $\mathcal{C}_k^i$ be the $i^{th}$ cluster $\in G_k$. As depicted in Fig. 6.7, the ontological class $l$ within $\mathcal{C}_k^i$ shall be denoted as $c_k^{il}$.

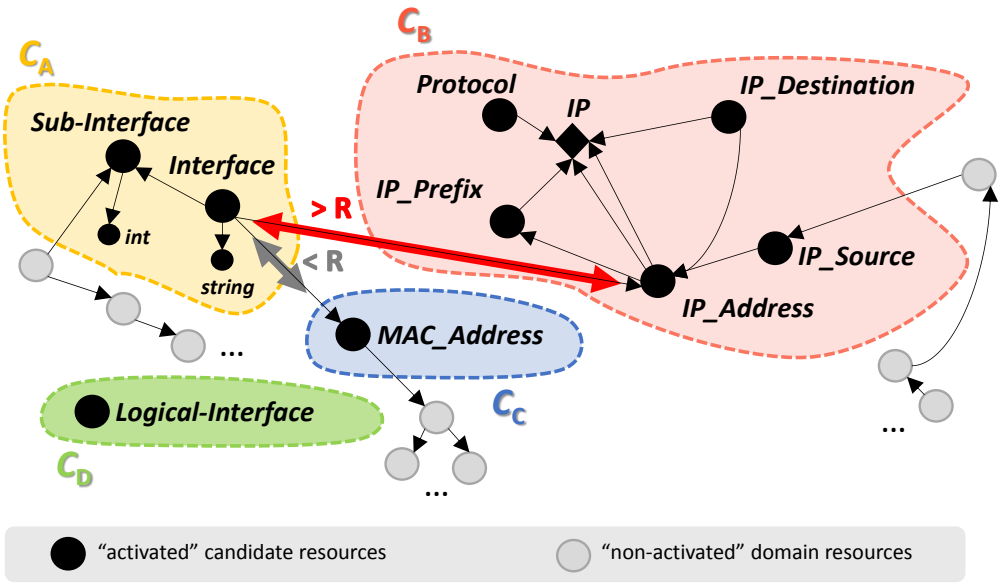


FIGURE 6.6: The rationale behind the quantification of the Semantic Relatedness.

*Definition* 2. The **Semantic Relatedness** ($\mathcal{R}$) between the set of concepts $c_k \in G_k$ is a means to determine the degree to which candidate resources are associated by meaning. It is defined as the product of the density ($d$) and the Information Content ($I$) (cf., Equation 6.1). As the majority of approaches we consider semantic relatedness symmetric.

$$\mathcal{R}(G_k) = d(G_k) \cdot \mathcal{I}(G_k) \tag{6.1}$$

Next, we will formally define the components for the computation of semantic relatedness, namely, **density** ($d$) and **Information Content** ($I$).

*Definition* 3. The graph interconnectivity which is captured under the notion of **density** ($d$) (cf., Equation 6.2) is computed as the relation between the number of *active* edges along the shortest path between any pair of clusters in graph $G_k$, and the total number of edges in those shortest paths (i.e., *path length*).

$$d(G_k) = \frac{\sum_{i=1}^{|\mathcal{C}_k|-1} \sum_{j=i+1}^{|\mathcal{C}_k|} [\mathcal{A}(\mathcal{SP}(\mathcal{C}_k^i, \mathcal{C}_k^j)) - 1]}{\sum_{i=1}^{|\mathcal{C}_k|-1} \sum_{j=i+1}^{|\mathcal{C}_k|} \mathcal{H}(\mathcal{SP}(\mathcal{C}_k^i, \mathcal{C}_k^j))} \leq 1 \tag{6.2}$$

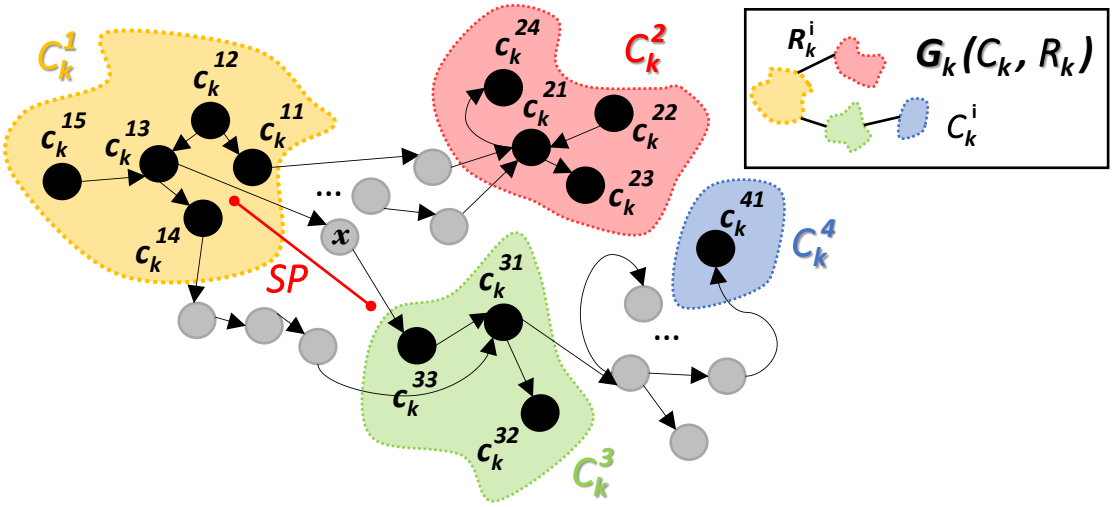


FIGURE 6.7: An example of Semantic Relatedness.

| | |
|---|---|
| $\mathcal{A}(\mathcal{P})$ | a function that returns the number of activated entities along a given *path* $P$. |
| $\mathcal{H}(\mathcal{P})$ | a function that returns the length of a given *path* $P$. |
| $\mathcal{SP}(\mathcal{C}_k^i, \mathcal{C}_k^j)$ | a function that returns the *shortest path* between a given pair of clusters. |

Let $\mathcal{C}_k^i$ and $\mathcal{C}_k^j$ be a pair of clusters in $G_k$, and let $\mathcal{P}(c_k^{il}, c_k^{jp})$ denote a path between a pair of entities $c_k^{il} \in \mathcal{C}_k^i$, and $c_k^{jp} \in \mathcal{C}_k^j$. The shortest path between two clusters is defined as $\mathcal{SP}(\mathcal{C}_k^i, \mathcal{C}_k^j) = \min\ \mathcal{P}(c_k^{il}, c_k^{jp})$, $\forall\ c_k^{il}, c_k^{jp}$ in clusters $\mathcal{C}_k^i$, and $\mathcal{C}_k^j$, respectively. The number of *active* edges is computed as the total number of activated entities along the path $(\mathcal{A}(\mathcal{P}) - 1)$. Recall that clusters are composed of activated entities only, thus, source and destination of any given path $P$ are always activated entities, hence the number of *active* edges is $(\mathcal{A}(\mathcal{P}) - 1)$. Notice that, similarly to the approach of Tsatsaronis et al. in [**?**] a *path* $(\mathcal{P})$ can be a combination of different types of edges—including both taxonomic and non-taxonomic. However, we consider that relations are equally weighted.

To illustrate this, consider the paths between the clusters $\mathcal{C}_k^1$ and $\mathcal{C}_k^3$ as shown in Fig. 6.7. In this case, the shortest path between any pair of entities $(c_k^{1l}, c_k^{3p})$, *i.e.*, paths with source in cluster $\mathcal{C}_k^1$ and termination in $\mathcal{C}_k^3$, or vice-versa, is $\mathcal{SP}(\mathcal{C}_k^1, \mathcal{C}_k^3) = [(c_k^{13}, x), (x, c_k^{33})]$. Now, let the function $\mathcal{A}(\mathcal{P})$ return the total number of "activated" entities (*i.e.*, the ontological classes) in path $\mathcal{P}$. In our example, $\mathcal{A}(\mathcal{SP}(\mathcal{C}_k^1, \mathcal{C}_k^3)) = 2$, which are $c_k^{13}$ and $c_k^{33}$. The function $\mathcal{H}(\mathcal{P})$ in the denominator of (Eq.6.2) returns the total number of hops in path $\mathcal{P}$. For instance, in the example shown in Fig. 6.7, $\mathcal{H}(\mathcal{SP}(\mathcal{C}_k^1, \mathcal{C}_k^3)) = 2$. Observe that when the clusters $\mathcal{C}_k^i$ and $\mathcal{C}_k^j$ are not adjacent, the shortest path can traverse other clusters. Hence, in a connected graph, the number of activated entities always satisfies $\mathcal{A}(\mathcal{SP}(\mathcal{C}_k^i, \mathcal{C}_k^j)) \geq 2$.

*Definition* 4. The **Information Content** $(I)$ is a measure of the knowledge enclosed by a cluster (cf. Eq. 6.3).

$$\mathcal{I}(G_k) = \sum_{i=1}^{|\mathcal{C}_k|} \sum_{l=1}^{|c_k^{il}|} t_k^{il}.m_k^{il}.o_k^{il} \tag{6.3}$$

In formulae, we use $t_k^{il}$ to denote the total number of terms that trigger the activation of a domain entity $c_k^{il} \in \mathcal{C}_k^i$ in the semantic graph $G_k$. Moreover, $m_k^{il}$ is a matching factor which represents the probability of an entity of being the asserted concept with respect to the total number of entities identified for the same set of terms. This coefficient takes the

maximum value of "1" whenever a domain entity has been identified by perfect lexical match, or $\frac{1}{(e+1)}$ in all other cases, with $e$ the total number of entities also identified for the same terms. In the example shown in Fig. 6.6, e = 2 for the entities triggered by the term "address", with equal probability from the information content perspective of being *"IP_Address"*, *"MAC_Address"*, or none of them. Finally, the $o_k^{il}$ factor is calculated by counting the frequency of occurrence of an entity for all levels, over the total number of occurrences of its exclusive disjoint candidate entities.

After computing semantic relatedness for all set of candidate concepts ($\forall G_k$), we select the set with maximum relatedness (cf., Eq. 6.4) as the most suitable set of resources representative of the CLIs knowledge. If semantic relatedness is the same for more than one set of clusters we compute out-degree as tie break.

$$\max_k \ \mathcal{R}(G_k) \tag{6.4}$$

It is worth mentioning that, even though at first sight our model might look a bit intricate, its computation is actually quite straightforward. The nature and hierarchical structure of CLIs typically yields a small number of interrelated clusters, and more importantly, as outlined in Fig. 5.1, this subsystem operates in offline mode, so the only and fundamental goal is the accuracy of the OBIE process. Indeed, the results that we present in Chapter 8 confirm the strengths of our model and the approach herein proposed.

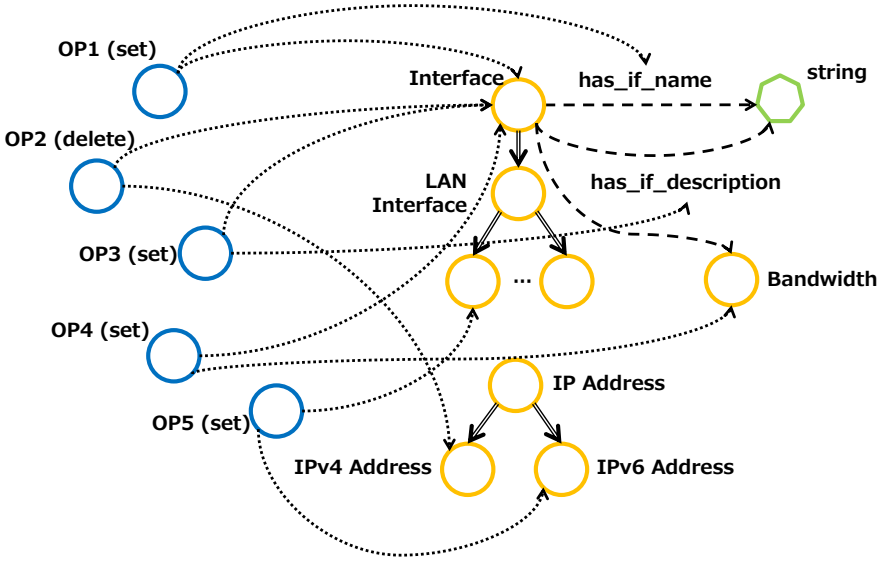


FIGURE 6.8: An example Decision Maker Stage.

## 6.3 Stage II: Operation Identification

### 6.3.1 Decision Maker and Ontology Population

In this stage we combine the information of identified network resources and verb phrases to semantically derive the operation(s) that a command or set of commands represent—i.e., with respect to components defined within the operation layer of our domain ontology. In other words, we reason over the ontology to determine the set of operations that most likely represent the semantics of a given configuration statement. If a single atomic operation does not fully-define the extracted information (i.e., network resources and verb phrases at the same time), we build semantic connections between atomic operations, which furthermore represent the way in which commands are organized in the hierarchy. Consider for instance the example shown in Fig. 6.8, where the set of identified resources are interface (L2), if-name (L3), if-description (L4) and verb phrases ***set*** (L1). The nature of our decision maker is to select on a level-basis the most concrete atomic operation and finally build a semantic flow among them. Therefore, in our example the output of this stage will be the concatenation of atomic operations OP1 - OP3. Under this scenario, whenever a user requests the semantic operation OP3, we will be able to automatically build the sequence of commands by following the path of semantic links. Ontology population is the actual instantiation of commands into the semantic categories. In the case that a single atomic operation is represented by several commands or combination of commmands and variables we include ordering information for adequately retrieving configuration statements in the exact order.

## 6.4 Extending the OBIE System to other Application Domains

One of the main features of the proposed architecture is to enable future portability of our system to other application domains, wherein configuration is also distributed, heterogeneous, command-based, and most importantly, hierarchical. Consider for instance, applying the system for the configuration of printers in a large business organization, the initial setup of multi-technology voting machines in an electoral process or the configuration of distributed medical equipment in a health facility.

Notice that, our methodology for Information Extraction from the CLI is independent of the conceptualizations of the underlying domain (in our case, L2/L3 network device configuration knowledge) which is actually externally provided by the domain ontology

(i.e., ONDC)—yet not specific to the learning algorithm. Instead, our methodology depends on features that are general (i.e., common) to *any* CLI-based environment, such as, implicit knowledge derived from the hierarchy or CLI structural parts (i.e., information given in the form of *commands*, *variables* and *help descriptors*). Accordingly, the requirements for extensibility of our system to other applications domains are restricted to the following.

*Extensibility Requirement* 1. **Develop and create an OWL ontology for the underlying domain of interest.**

In the context of our approach, the knowledge of the domain is externally provided by an ontology which formally specifies the concepts and relations of the domain of interest. Indeed, the ontology plays a crucial role as it guides the process of Information Extraction. For this reason, the definition of such knowledge model must satisfy certain guidelines which are given below.

- The ontology must be specified in the W3C OWL Web Ontology Language.
- It must consider all OWL language constructs, namely, classes, data and object properties, individuals, restrictions and pattern facets.
- The domain lexicon must be integrated into the ontology. To this end, OWL classes must be annotated with meta-data which provides alternative $\langle keywords \rangle$ (i.e., synonyms) for the referred concepts.
- The ontology must model both *taxonomic* and *non-taxonomic* relations between concepts. Regarding *non-taxonomic* relations we will confine the knowledge model to *meronomy* semantic relations, i.e., part-whole ("*has-a*") relation types.
- Acronyms must be labeled with the $\langle acronym \rangle$ meta-data type.
- It is mandatory to define the ontology in two layers, namely, a *resource* layer and an *operation* layer, as previously described in Section 5.3.2.
- OWL classes within the *operation* layer must be enriched with the $\langle verb \rangle$ meta-data type, indicating the verb(s) that best suit the semantics of the atomic configuration action.
- Domain data-types must be modeled using the pattern facet restriction feature of OWL2 in order to match with regular expressions.

*Extensibility Requirement* 2. **Provide the device CLI as input to the system.**

Despite the heterogeneity of CLI-based environments regarding semantics (i.e., meaning), syntax (i.e., terminologies) and hierarchy granularity, they typically share a set of features which must be verified to ensure system compatibility:

- CLIs must be hierarchical and relational.
- CLIs must be largely based on Natural Language Text.
- CLIs must be structured in the form of *commands* and *variables* with corresponding *help descriptors*.

*Extensibility Requirement* 3. **Provide list of domain-related verbs.**

The system requires an external list of the verbs that most likely define the actions of the underlying domain. Notice that, because the potential fields of application of our system are related to the "configuration" of some type of device, this list can be considered the same across all domains. Nevertheless, in the case that new verbs are required functions are provided by the *Administration Controller*.

In short, the proposed architecture ensures extensibility to other application domains. For adaptation, users must complete the aforementioned requirements.

# Chapter 7

# Command Retrieval Algorithm

As stated earlier, the *online* functional mode supports the semantic retrieval of configuration statements. It semantically resolves configuration requests on the basis of heterogeneity and automatically retrieves the sequence of commands for a given device model. It provides a web-based interface for third-party applications willing to outsource the task of configuration adaptation to our system. Moreover, the online process supports advanced functions that enable format adaptation for well-known domain concepts. For example, it automatically performs subnet format adaptation. Notice that this knowledge is embedded in the domain ontology and adaptation functions are automatically triggered whenever an input differs from the expected type.

To illustrate the online functionality, consider the following use case. An NMS that targets network programmability is configured to automatically offload the traffic of a gold client over a new path whenever the traffic in the primary path goes beyond a certain threshold. In order to achieve this, current solutions rely on manually set configuration scripts—a strategy commonly used to simplify recurrent tasks—which depends on the underlying infrastructure. However, a solution of this type can actually outsource the configuration of the devices involved to our OBIE system, and request the required configuration in runtime —regardless of the underlying infrastructure— thus, decoupling network programmability from the specifics of the current network setting. Therefore, any change in the network (*e.g.*, new devices, operating system upgrades, etc.) will not affect regular ongoing processes. The online functionality has already been developed, tested and successfully validated in the framework of an European initiative, enabling management programmability in the context of multi-layer and multi-vendor networks [111].

# Chapter 8

# Experiments and System Validation

*This Chapter presents the performance evaluation of our Ontology-Based Information Extraction System from the CLI of network devices. We further investigate the impact of various factors on the performance of our system.*

## 8.1 Experimental Framework

The system was entirely developed in the Java Programming Language as it ideally suits the needs of integration with available libraries for OWL Ontology Management (OWL API [185]) and Natural Language Processing (NLP Stanford [187]). Furthermore, we created the switch/router configuration domain ontology in the Protégé Editor based on the Web Ontology Language (OWL). This ontology is the result of the consensus of experts in the networking area and contains a representation of over 600 resources and near 320 operations. The **O**ntology for the **N**etworking **D**evice **C**onfiguration (ONDC) domain can be found in [165] for download.

In order to assess the performance of our system we carried out end-to-end experiments over the configuration spaces of well-known routing devices—both proprietary and open-source. Most specifically, we performed evaluations over the configuration spaces of a Juniper Router (Model M7i - JUNOS 10.4.R13.4), a Cisco Router (Cisco IOS Release Version 12.4(16a) FC2) and the Quagga Routing Software (Release 0.99.21). We manually limited each CLI to a set of commands which represent—from a semantic perspective—the operations most commonly performed by network administrators

across their networks. Notice that, though the number of commands for a network device can be significantly large, the ones actually used in practice are a relative small set. In light of this, operations were determined by thoroughly analyzing the configuration files of actual deployed core routers. We selected around a total of 300 atomic operations, which individually map to heterogeneous sets of commands in each of the selected configuration spaces. It is important to realize that the downsized CLI encompasses a broad set of functionalities—i.e., not only protocol-specific configurations (e.g., OSPF, SNMP) but device-related functions as well (e.g., user account settings).

The final step is to actually perform the off-line execution of our OBIE algorithm over the restricted CLI environments for each of the three device models. Next, we will provide some insights on the most common metrics for OBIE system performance evaluation.

### 8.1.1 Evaluation Metrics

In the literature, ***Precision*** (P), ***Recall*** (R) and ***F-measure*** (F) have become the absolute metrics for the performance evaluation of traditional Information Extraction Systems [189]. Overall, these measures reveal the ability of a system to identify information from text. Most specifically, *precision* is a measure of correctness—i.e., a percentage of correct instances with respect to the total number of identified items. *Recall* is a measure of completeness—i.e., a percentage of correct instances with respect to the total number of expected identified instances. Finally, *F-measure* is the geometric mean of both metrics, and reflects the overall quality of the instantiations. However, when Information Extraction is done with respect to components of an ontology, traditional metrics are insufficient [190]. This is mainly because conventional measures have a binary behavior when determining the correctness of an instantiation. While in the context of ontological classification, decisions are more obscure, i.e., we can more or less have different "degrees" of correctness. For instance, consider classifying the instance 'FE80::0202:B3FF:FE1E:8329' as an $\langle IP\_Address \rangle$ rather than an $\langle IPv6\_Address \rangle$—which is in fact a more accurate classification. In this case, classification is clearly less wrong than if it was classified as an instance of the $\langle MAC\_Address \rangle$ class. Accordingly, we require of other mechanisms which provide the means to quantify the degree of correctness of instantiations. One of the most common used metrics for the evaluation of OBIE systems is the Balanced Distance Metric (BDM) proposed by authors in [190]—a cost-based component that measures the degree of correctness according to the ontological structure. In order to compute this component we used the open-source BDM Computation Processing Resource, which is part of the General Architecture for Text Engineering (GATE) platform [191]. This tool outputs a file with the BDM scores for all pair of classes in the ontology. Notice that, the BDM by itself, does not provide the

means for evaluation of the system's overall performance. For this reason, we computed augmented versions of traditional Precision, Recall and F-measure as defined by the same authors in [192] (cf., Equation 8.1).

$$AP = \frac{BDM}{n + Spurious} \qquad AR = \frac{BDM}{n + Missing} \qquad AF = \frac{AP \cdot AR}{0.5 \cdot (AP + AR)} \tag{8.1}$$

In order to compute performance metrics, we manually built a *gold standard* for each configuration space. The *gold standard* represents a benchmark data set against which to compare the system's output. It was created by a group of networking experts as a fully-compliant reference set of the most adequate semantic annotations from the domain ontology. We created a gold standard for each stage of our instantiation methodology, namely, one for the resource identification stage (**Stage I**) and another for the operation identification stage (**Stage II**). Results for **Stage I** are a measure of the ability of our system to extract information of networking resources from the CLI, while results for **Stage II** measure the ability of our system to derive and infer the configuration operations that commands in the CLI actually represent. The main motivation for reporting performance on both stages is to test the success of a two-fold approach for IE. We then compare extractions with respect to the gold standard and compute augmented performance metrics.

## 8.2 Performance Results

In this section, we will report the performance results of our system. From now on, we will refer to the initial version of our IE algorithm (as earlier described in Chapter 6) as the Command Line Information Extraction (*CLIE*) Algorithm. Based on the observed results, we will later motivate algorithmic changes and investigate the impact on the performance of our system.

### 8.2.1 *CLIE* Algorithm

The performance results of our system for the *CLIE* Algorithm are shown in Table 8.1. This table depicts the final percentage values for Augmented Precision (AP), Recall (AR) and F1-Measure (AF), as well as conventional values for the same metrics (P, R and F1) for both stages of our algorithm, separately. Overall, experimental results show that our system is capable of automatically extracting the semantics of the configuration

environment from the CLI with high performance. The system achieves around 91% AP and 91% AR, which is near human-level performance.

| | **Augmented** | | | | | | **Traditional** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ***Stage I* (%)** | | | ***Stage II* (%)** | | | ***Stage I* (%)** | | | ***Stage II* (%)** | | |
| | **AP** | **AR** | **AF** | **AP** | **AR** | **AF** | **P** | **R** | **F** | **P** | **R** | **F** |
| *Juniper* | 83.7 | 93.4 | 88.3 | 94.0 | 94.0 | 94.0 | 80.6 | 90.0 | 85.0 | 89.9 | 89.9 | 89.9 |
| *Cisco* | 86.0 | 93.3 | 89.9 | 90.6 | 90.6 | 90.6 | 85.6 | 92.9 | 89.1 | 87.5 | 87.5 | 87.5 |
| *Quagga* | 88.0 | 86.6 | 87.3 | 88.1 | 87.9 | 88.0 | 87.2 | 85.8 | 86.5 | 83.0 | 82.8 | 82.9 |
| **Overall** | 85.9 | 91.0 | 88.4 | **90.8** | **90.8** | **90.8** | 84.5 | 89.6 | 87.0 | 86.8 | 86.7 | 86.7 |

TABLE 8.1: Performance Results OBIE Process (Augmented and Traditional): *CLIE* Algorithm.

When comparing augmented values against traditional metrics of P, R and F1-Measure (cf., Table 8.1) we can confirm that our algorithm has the ability to approximate concepts whenever the information given in the CLI is not sufficient to unambiguously determine its semantics. However, the fact that both metrics do not *significantly* differ allow us to conclude that—in general terms—semantic categorization is highly asserted and not primarily dependent on approximations.

Notice that with 91% of AP and AR achieved in a fully automated matter, near 9% of commands would not be adequately instantiated within their corresponding semantic categories. Nevertheless, considering that significant tedious work of mapping would be done and that we can have good suggestions produced by the system, it is reasonable to think of a human in the final loop of deployment. Indeed, a network administrator would have a significantly easier time verifying the remaining 9% of commands than navigating and semantically interpreting the entire hierarchy.

Moreover, we identified an additional capability of our system in determining correspondence (i.e., consistency) between the facts expressed in the CLI and general networking knowledge—i.e., knowledge as reflected in the domain ontology (ONDC). An analysis of the experimental results show that whenever the CLI is inconsistent with respect to the networking literature—i.e., not strictly aligned to the domain knowledge—our system approximates (either by generalizing or specifying) to the nearest concept, wherein these assertions are true. To illustrate this, let us consider the following example. A given CLI arranges the "MPLS"-related commands under the "Routing Protocol" level of the hierarchy. However, it is a fact that in the literature MPLS is not actually considered a Routing Protocol. In light of this, although our system identifies the

$\langle Routing\_Protocol \rangle$ concept in the early stage of lexical matching (as expected), in further stages of inference and clustering—i.e., when making associations between concepts in contiguous levels—it generalizes it to the $\langle Standard\_Protocol \rangle$ concept. The reason for this is that according to the ontology axioms, the $\langle MPLS \rangle$ concept identified in subsequent levels is not of the type "Routing Protocol". This means that, most likely $\langle MPLS \rangle$ and $\langle Routing\_Protocol \rangle$ cannot be at the same time candidates for the same configuration statement. Given that the taxonomic ancestor of $\langle Routing\_Protocol \rangle$ , (*i.e.*, $\langle Standard\_Protocol \rangle$) is semantically related to $\langle MPLS \rangle$ we perform concept generalization. Notice that this example is not actually a problem of miss-classification, but rather a problem derived from an inconsistency between the literature and the vendor's interpretation of the domain's knowledge. Thus, we can conclude that the generalization feature of our system can certainly help to identify and bridge the gap between the CLI information and the domain knowledge model, whenever inconsistencies take place. These exceptions are out of the scope of this work, and will be considered in future work.

Furthermore, in an effort to assess the suitability of our design decisions, we developed potential improved versions of our IE algorithm by taking into account other variables in the decision process. The *CLIE* version of our IE algorithm computes semantic association between candidate concepts based on the relatedness measure ($\mathcal{R}$) defined in Equation 6.1. One of the potential paths of enhancement that we performed—and which has been extensively explored in other research areas—was considering the ontology *depth* as a variable to the computation of semantic relatedness. The notion of including ontology depth as a new variable for semantic relatedness computation is gathered under the concept of the Depth Command Line Information Extraction *D-CLIE* Algorithm.

### 8.2.2 *D-CLIE* Algorithm

Ontological depth is one of many variables considered in several state-of-the-art semantic relatedness metrics. It is based on the notion that deeper nodes in the taxonomy—i.e., more specific concepts—have stronger semantic association than higher (generic) concepts [156]. In light of this, we included a component to our relatedness measure which considers the relative depth of clusters (denoted as $\mathcal{D}$) (cf., Eq. 8.2 and 8.3). This component is a relation of the sum of actual depths of a pair of clusters over the sum of the maximum depths. We refer to the version of our algorithm which considers ontology depth as a variable for relatedness computation as the Depth Command Line Information Extraction (*D-CLIE*) Algorithm.

$$d'(G_k) = d(G_k) \cdot \sum_{i=1}^{|\mathcal{C}_k|-1} \sum_{j=i+1}^{|\mathcal{C}_k|} \mathcal{D}(\mathcal{C}_i, \mathcal{C}_j) \tag{8.2}$$

$$\mathcal{D}(\mathcal{C}_i, \mathcal{C}_j) = \frac{depth(\mathcal{C}_i) + depth(\mathcal{C}_j)}{max\_depth(\mathcal{C}_i) + max\_depth(\mathcal{C}_j)} \tag{8.3}$$

| | |
|---|---|
| $depth(\mathcal{C}_n)$ | a function that returns the depth of a cluster as the depth of the deepest node in the cluster. |
| $max\_depth(\mathcal{C}_n)$ | a function that returns the maximum depth of a cluster computed as the deepest concept among all branches of the cluster. |

The performance results of our system for the *D-CLIE* Algorithm are depicted in Table 8.2. Notice that overall, the system's performance is not affected by the inclusion of a new variable to our semantic relatedness measure—when metrics are compared to those obtained for the initial version of our algorithm (cf., Table 8.1). We believe that in the context of our approach, the consideration of ontology depth as a variable for semantic relatedness computation has no significant impact because of the generalization feature performed during lexical matching and the subsequent reasoning in the inference stage. Accordingly, concepts within a branch of the ontology (i.e., taxonomically related concepts) have already been pruned on this basis. As such, by the time we compute semantic relatedness, pairs of disjoint candidate concepts on a same taxonomy branch do not exist, therefore, ontology depth has no actual weight in the final scores. For this reason, we performed new experiments in which we removed the generalization and specialization features of our algorithm in order to do taxonomic pruning based on semantic relatedness measures which take into account depth (*D-CLIE*$^*$ Algorithm). The main motivation is to determine whether generalization and inference show better results for taxonomic pruning than a strategy of semantic relatedness computation based on ontology depth consideration.

### 8.2.3 *D-CLIE*$^*$ Algorithm

The results of this experiment are shown in Table 8.3. Notice that, though the system's performance does not dramatically decrease, it is certainly lower than the performance

| | ***Stage 1*** **(%)** | | | ***Stage 2*** **(%)** | | |
|---|---|---|---|---|---|---|
| | **AP** | **AR** | **AF** | **AP** | **AR** | **AF** |
| *Juniper* | 83.7 | 93.4 | 88.3 | 94.0 | 94.0 | 94.0 |
| *Cisco* | 85.9 | 93.1 | 89.4 | 90.6 | 90.6 | 90.6 |
| *Quagga* | 89.0 | 86.6 | 87.3 | 88.1 | 87.9 | 88.0 |
| **Overall** | 85.9 | 90.9 | 88.3 | **90.8** | **90.8** | **90.8** |

TABLE 8.2: Performance Results OBIE Process: *D-CLIE* Algorithm.

metrics obtained for the initial *CLIE* Algorithm (cf., Table 8.1). Accordingly, considering the ontology depth as the absolute criteria for taxonomic pruning of concepts does not improve the performance of the system. We strongly believe that ontology depth does not compensate to the same level to the performance of our generalization and reasoning features, basically because our decisions rely on the lexical knowledge obtained from the CLI in combination with ontological structure and relevant ontological facts, such as non-taxonomic relations. While ontology depth only takes a decision based on the taxonomy. In light of all this, we can definitely conclude on the suitability of our initial design premises as the system achieves the highest performance values for this scenario.

| | ***D-CLIE**** | | | | | |
|---|---|---|---|---|---|---|
| | ***Stage I*** **(%)** | | | ***Stage II*** **(%)** | | |
| | **AP** | **AR** | **AF** | **AP** | **AR** | **AF** |
| *Juniper* | 75.4 | 89.1 | 81.7 | 89.2 | 89.2 | 89.2 |
| *Cisco* | 80.1 | 87.4 | 83.6 | 86.7 | 86.7 | 86.7 |
| *Quagga* | 84.0 | 81.6 | 82.8 | 85.0 | 84.8 | 84.9 |
| **Overall** | 79.8 | 86.0 | 82.8 | **86.8** | **86.8** | **86.8** |

TABLE 8.3: Performance Results OBIE Process: *D-CLIE** Algorithm.

### 8.2.4 Tsatsaronis et al. Relatedness Metric

In an effort to validate the suitability of a CLI feature-dependent metric, we replaced our own semantic relatedness measure with that proposed by authors in [193]. The main

motivation is to assess the performance of our system when considering a non CLI-dependent relatedness measure—still closely aligned to our research goals. Notice that, all other steps of our Information Extraction Algorithm, namely, data pre-processing, lexical matching, clustering, inference and decision making remain invariant for this validation scenario.

Tsatsaronis et al. [193] propose a metric for semantic relatedness computation (cf., Eq. 8.4) in which the weighted path *length* and actual path *depth* between concepts in an ontological graph are considered. While *path length* is captured under the notion of *compactness* (cf., Eq. 8.5), *path depth* is captured under the concept of semantic path elaboration (*spe*) (cf., Eq. 8.6). Similar to our approach, authors consider both taxonomic and non-taxonomic relations, while giving different weights ($w$) to each type. The way in which path *length* and *depth* is computed certainly differs from our approach, but most importantly, there is no particular consideration of CLI-dependent features—which in our case is captured under the notion of the Information Content (IC) component. Equation 8.4 represents the Relatedness measure proposed by authors as the maximum product of *compactness* and *spe* given by any path between them. Notice that, among the types of relations described by authors in [193] we restricted types to the following, hypernym/hyponym ($w = 0.61$), part meronym/holonym ($w = 0.0367$) and attribute ($w = 0.00414$), with the same weight values considered by authors.

$$\mathcal{R}(c_1, c_2) = \mathbf{max}(compactness(p(c_1, c_2)) \cdot spe(p(c_1, c_2))) \tag{8.4}$$

$$compactness(p(c_1, c_2)) = \prod_{i}^{l} w(e_i) \tag{8.5}$$

$$spe(p(c_1, c_2)) = \prod_{i}^{l} \frac{2 \cdot depth(c'_i) \cdot depth(c'_{i+1})}{depth(c'_i) + depth(c'_{i+1})} \cdot \frac{1}{depth(T)} \tag{8.6}$$

The results of executing our IE algorithm with semantic relatedness based on the external metric proposed by Tsatsaronis et al. are shown in Table 8.4. Notice that overall, system's performance drops 9% with respect to our initial proposed *CLIE* Algorithm—i.e., from 90.8% to 81.9% AF. This means that, considering CLI-related features in the computation of semantic relatedness does improve accuracy in semantic instantiation for the case of CLI environments. It is important to realize that semantic relatedness computation directly affects Stage I of our algorithm, for which performance drops in

10%, meaning that, the correct identification of CLI concepts decreases and directly impacts on the decision stage of our algorithm (i.e., Stage II).

| | ***Tsatsaronis et al.*** | | | | | |
|---|---|---|---|---|---|---|
| | ***Stage I*** **(%)** | | | ***Stage II*** **(%)** | | |
| | **AP** | **AR** | **AF** | **AP** | **AR** | **AF** |
| *Juniper* | 73.9 | 80.2 | 76.9 | 81.8 | 81.8 | 81.8 |
| *Cisco* | 76.2 | 79.9 | 78.0 | 83.6 | 83.6 | 83.6 |
| *Quagga* | 79.3 | 81.3 | 80.3 | 82.6 | 82.5 | 82.6 |
| **Overall** | 76.4 | 80.4 | 78.3 | **82.0** | **81.8** | **81.9** |

TABLE 8.4: Performance Results OBIE Process: Tsatsaronis et al. Relatedness Measure

## 8.3 Enhancing CLIs with Meta-Data

The information provided in CLI-based environments frequently lacks of verbosity, a fact which can certainly lead to ambiguity or semantic incompleteness. In light of this, it seems reasonable to think that by enhancing CLIs with meta-data, configuration knowledge extraction could be improved. However, current CLIs are not editable and thus, there are no means to augment the semantics of the help feature. Based on this motivation, we will briefly describe an use case of the OPENER tool to enable potential edition of CLIs.

### 8.3.1 OPENER

#### 8.3.1.1 In a Nutshell

Software Defined Networking (SDN) offers endless opportunities for creative researchers and entrepreneurs, and confers greater agility to the networking industry to meet the flexibility and cost reductions that service providers are constantly seeking. This paradigm shift toward greater openness to the software industry is increasingly attracting researchers and developers, and it is fostering innovations in the networking arena at an

astonishing speed. In spite of this, researchers with the capacity to devise really creative SDN applications lack a non-proprietary environment in which they can easily manage and experiment with their creations. This is especially the case when the targeted SDN applications have requirements that go beyond the OpenFlow philosophy. In this context, we introduce OPENER as a tool that offers an open and programmable environment that covers the entire life-cycle for managing the experimentation with out-of-the-box SDN applications. OPENER can be installed and used in a standalone fashion, providing the development, management, and execution environments facilitating the implementation, deployment, and testing of SDN applications beyond what non-proprietary solutions currently offer. Moreover, OPENER depends neither on specific SDN platforms nor on particular APIs and protocols, which makes it highly flexible for application developers.

#### 8.3.1.2 The Configurable Helps Use Case

One of the potential use cases of the OPENER platform is developing an SDN-based application to modify and enhance the CLIs help feature of Quagga Routers—a function which unfortunately is not provided at present by any vendor in the market. A solution of this nature would provide the functions to augment the semantics of CLIs help descriptors, and furthermore—in the context of our solution—allow us to analyze whether such strategy improves our systems performance or not, i.e., improve the ability to interpret and derive the semantics of CLI environments. If successful, we could derive information models for the CLI helps, and make recommendations to device vendors for restructuring and adding the proper semantics to their CLI helps.

To tackle this limitation, we have seized the functionalities provided by OPENER. For this purpose, first, we made Quagga SDN-enabled, by implementing wrapper functions able to expose its internal functionalities, independently of OPENER. Then, we developed the SDN Technology-Dependent Abstraction (STDA) to bind the OPENER platform independent features to the particularities of our SDN-enabled Quagga. Finally, we developed an OPENER application that hooks itself to the core of the Command-Line system, overriding its default internal mechanisms through a callback, which decodes the original CLI helps and commands, and allows their modification. Using this approach, the system can always revert to legacy mode by removing these wrappers, leaving Quagga with its original features. This application enables the edition of the CLIs command help descriptors, and augmenting its semantics through the addition of meta-information. Figure 8.1 shows this application in action. It is built on top of the already available command-line user interface, and it enables a set of new CLI commands for creating, editing, appending, saving and retrieving new descriptions and

meta-information to any existing command. The example shown in Fig. 8.1 illustrates how the help descriptor for the command "show ip" can be easily modified. Table 8.5 summarizes the CLI functions that have been developed on OPENER to endow the configuration environment with richer semantics for experimentation.

Note that this use case aims at investigating and experimenting with functions that are far beyond the OpenFlow philosophy and its capabilities, though the use case is

```
User Access Verification

Password:
Router>
Router>
Router> show
  debugging     Zebra configuration
  history       Display the session command history
  interface     Interface status and configuration
  ip            IP information
  ipv6          IPv6 information
  logging       Show current logging configuration
  memory        Memory statistics
  table         default routing table to use for all clients
  thread        Thread information
  version       Displays zebra version
  work-queues   Work Queue information
```

(a) Default help supplied by a Quagga router for the "show ip" command.

```
Router> help create Detailed IPv4 information including: access-list, forwarding, prefix-list and route.
> Internal data is now:  "Detailed IPv4 information including: access-list, forwarding, prefix-list and route."
Router>
Router>
Router> help set show ip
> New struct Metadata created
> Command description modified
Router>
Router>
Router>
Router> show
  debugging     Zebra configuration
  history       Display the session command history
  interface     Interface status and configuration
  ip            Detailed IPv4 information including: access-list, forwarding, prefix-list and route.
  ipv6          IPv6 information
  logging       Show current logging configuration
```

(b) Modifying the help descriptor for the "show ip" command using the new commands devised "help create" and "help set".

```
Router> metainfo create <opener> <view> <helpModified>
> Internal data is now:  "<opener> <view> <helpModified>"
Router>
Router> metainfo set show ip
> Metadata modified
Router>
Router>
Router>
Router> show
  debugging     Zebra configuration
  history       Display the session command history
  interface     Interface status and configuration
  ip            Detailed IPv4 information including: access-list, forwarding, prefix-list and route.  MI:<<opener> <v
iew> <helpModified>>
  ipv6          IPv6 information
  logging       Show current logging configuration
```

(c) Example showing how to add Meta-Information (MI) to the "show ip" command using the new commands "metainfo create" and "metainfo set".

FIGURE 8.1: An SDN application on OPENER for endowing routers with configurable CLI helps.

| Command | Description |
|---|---|
| **metainfo/help *create*** | The *create* command reserves space in memory to store a string enclosing the customized help description or metadata. It can handle any size of string of data. |
| **metainfo/help *set*** | The *set* command allows to set the meta-information or help description of a given command to the newly created string data. This command overwrites the content. |
| **metainfo/help *append*** | The *append* command allows to append new data to the help descriptor or meta-info of a given command. |
| **metainfo/help *get*** | The *get* command retrieves the information for a given command. |
| **metainfo *clear*** | The *clear* command removes the metainformation for a given command. |
| **help *restore*** | The *restore* command allows to recover the default help description for a given command. |
| ***save* metadata** | The *save* command stores the modified help to a file. |
| ***load* metadata** | The *load* command allows to open the custom help file to start editing or visualizing its content. This command is automatically executed when booting the Quagga daemon. |

TABLE 8.5: New commands on a Quagga's CLI with OPENER implementation.

still perfectly aligned with the overall SDN paradigm. The central issue is that, rather than targeting the flows through the router, this use case targets the router itself. This use case offers a clear example of OPENER's potential for developing and managing non-flow based SDN applications.

### 8.3.2 Results

With support of the developed OPENER application we enhanced Quagga's CLI by adding additional meta-data to the help feature. This experiment demonstrates that the system's performance improves as additional information is provided to the learning algorithm in the form of meta-data. It can be seen from Table 8.6, that AF reaches 95.9% for the case of enhanced CLIs which is greater than the best result reported for our IE Algorithm. Moreover, notice that if we enrich the semantics of the CLI by adding semantic annotations from the developed ontology (as meta-data), maximum performance of our system can be achieved. However, this strategy is not feasible in the short term, neither because this would require standardization efforts toward the

ontological model nor because addition of meta-data is far from being an open feature in proprietary developments.

| | **CLIE Algorithm** | | | | | |
|---|---|---|---|---|---|---|
| | ***Stage I*** **(%)** | | | ***Stage II*** **(%)** | | |
| | **AP** | **AR** | **AF** | **AP** | **AR** | **AF** |
| *Quagga* | 88.0 | 86.6 | 87.3 | 88.1 | 87.9 | 88.0 |
| *Quagga + Meta-Data* | 94.2 | 93.2 | 93.7 | 95.2 | 96.7 | 95.9 |

TABLE 8.6: Performance Results OBIE Process: *CLIE* Algorithm + Augmented CLIs.

# Part IV

# SEMANTIC-BASED ADDRESSING

# Chapter 9

# The Current Internet

*In this Chapter we first provide the background on main limitations of the current Internet in the scope of routing and addressing with the aim of motivating the need of an Identifier/Locator Split Architecture. Afterwards, we describe the fundamentals of the proposed LISP Redundancy Protocol (LRP) and then introduce three critical scenarios in which LRP can assist to ensure reachability and reliability.*

## 9.1 Background

The current Internet architecture is organized into Autonomous Systems (ASes). The ASes interconnections generate a hierarchy between different Internet Service Providers (ISPs). In this hierarchical network structure, the Internet routing system is largely based in the Border Gateway Protocol (BGP) [194]—a long lived path-vector protocol which is used to exchange reachability information between ASes. Being a policy-routing protocol, it provides operators with the freedom to express their enterprise requirements and policies, allowing the attachment of several attributes for each route or network prefix. However, it has been largely demonstrated in the literature that the currently deployed Internet architecture suffers from serious weaknesses, mainly on the terms of routing scalability that along with the specific problems on the Internet addressing scheme require the deployment of new solutions. Next, we will provide insights on the most relevant aspects limiting routing performance on the overall Internet.

### 9.1.1 Internet Routing Scalability Problems

Recent studies including the Internet Architecture Board (IAB) report [4], reveal that Internet Routing is facing serious scalability problems, all involving both the size and

dynamics of the global routing table in the Internet's Default Free Zone (DFZ). The global routing table size in the DFZ has been growing at an alarming rate in recent years [195], until reaching now a total of 36.717 ASes that originate 355.262 IPv4 prefixes despite several limitations such as lack of IPv4 addresses, strict address allocation and routing announcement policies. Although IPv6 deployment would remove the problem of lack of IPv4 addresses, there is a strong concern that the deployment of IPv6 on a large scale could result in a significant growth of the routing table.

The IAB report [4] identified the following factors as the main reasons behind the rapid growth of the global routing table in the DFZ:

- Multi-homing.
- Traffic engineering.
- Non-aggregable address allocations.

In [196] authors conclude that address fragmentation, caused by multi-homing and load balancing is the major reason of BGP table growth.

### 9.1.2 Dynamics of the BGP Control Plane Information

In [197], a systematic study of highly active prefixes is presented, concluding that a small fraction of advertised prefixes are responsible for a relevant amount of churn in BGP; furthermore, they found that some generators of BGP beacons, used for active monitoring of BGP updates, appear as highly active. Despite the big amount of related work, the dynamics of the BGP control plane information (i.e., the exchanging of updates messages due to the advertisement of new prefixes) remains unknown, but certain evidence exists of Long Range Dependence [198]. As BGP propagates changes to the best path, a single router may send multiple updates based on one triggering event, and further, cause induced updates at other locations; examples of such events are link failures, newly added networks, prefix deaggregation and policy changes, among others. Moreover, it is important to notice that, since the routing information is subject to successive filtering by internal ASes policies, any route view of the network is always partial, determined by the local point of observation. On the other hand, a relatively small number of ASes are responsible for a disproportionately large fraction of the update churn that is observed today. In turn, another problem motivated by the growing of the BGP updates is the BGP convergence time, since as the larger the topology complexity is the longest the convergence time, hence motivating the network to take longer to recover from failures.

### 9.1.3 Multihoming Sites

Another factor related to the growth of the routing table refers to the multi-homing sites. For a network edge to be reachable by any service provider, the network-edge address-prefixes should be visible in the global routing tables. Meaning that no service provider can aggregate these address prefixes within their own address prefix, even if the network edges have addresses that belong to the provider-assigned address block. In addition, the network edges are increasingly obtaining provider-independent addresses from the Regional Internet Registries (RIRs), in order to avoid the renumbering every time a change of service provider happens. In summary, the topological information based on prefix-aggregation per provider is badly altered by multihoming, and in turn, leads to rapid growth of the global routing table.

### 9.1.4 Semantic Overloading

Another critical problem is the semantic overloading of IP addresses. This is because an IP address identifies a host (in fact its interface), and also serves to locate the host on the network. In this addressing scheme when a host changes of network provider, its IP address changes, therefore changing not only the network providing host access but also the host identifier. For upper layer applications that have IP addresses hard-coded for a host, this represents a severe mobility constraint. In short, the semantic overloading of IP addresses is the main problem when talking about renumbering a network.

## 9.2 The LISP Redundancy Protocol (LRP)

The LISP Control Plane proposed by authors in [25] aims to overcome the issues of current mapping systems, namely, ALT, CONS or NERD. The main issue of these approaches is related to the mapping resolution for a prefixed EID for the first outgoing packet—most frequently known as the "First Packet Drop Problem". Furthermore, this issue also leads to potential increase of network latency. The newly proposed control plane is based on the notion of retrieving EID-to-RLOC mappings within the Domain Name System (DNS) Resolution time. A major shortcoming of this solution is that mappings are replicated in all edge routers within the same AS. Despite the fact that an approach of such nature ensures improved reachability, unfortunately, it may raise serious scalability issues as each single router must store mapping information that rarely needs to use. As the mapping table size increases, the latency to find a mapping increases as well. This new issue directly affects the memory component within the

router, which is currently a bottleneck in the computer system when compared with processing capacity.

In an effort to minimize the mapping table size, the latency time, and hence, ensure the highest possible reachability, we propose the LISP Redundancy Protocol (LRP), which is inspired by the Cisco's Hot Standby Routing Protocol (HSRP) [199]. This architectural approach essentially allows to configure two routers for mapping purposes, so in case of single-failure, the other can replace it—following a Master-Slave model. The contribution of LRP is to extend HSRP functionalities by creating different logical groups. In this scenario, border routers can be members of different groups, and the "key" difference between LRP and HSRP focuses on the fact that LRP enables a router to be Slave in a group and Master in another (cf., Fig. 9.1). By implementing this feature all routers can be in forwarding mode, hence overcoming the limitation present in [25], namely to avoid the need of replicating the entire mapping on all the border routers. In case of either a link or border router failure, the last one will interchange his mapping with the border router that is now responsible of handling the traffic in this logical group.

In short, the main features offered by the LISP Redundancy Protocol are:

- The xTRs can be clustered into different LRP groups or pairs.
- The Mappings are pushed onto the LRP groups or pairs.

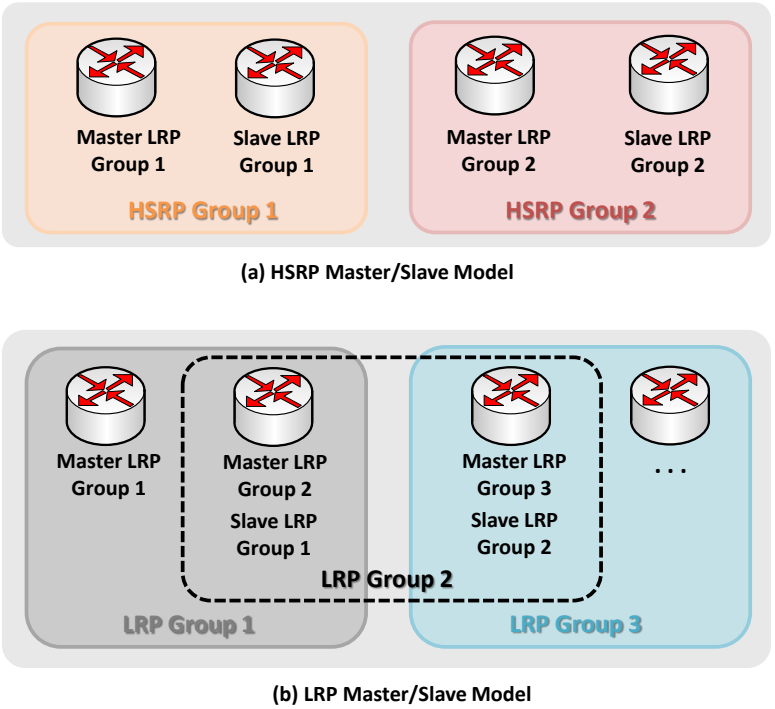


FIGURE 9.1: Master/Slave Model HSRP vs. LRP.

- All the xTRs in the group can carry traffic (active rather than standby).
- No need for data-probes when the xTR does not have a mapping.

Next, we consider different scenarios that may originate reachability and/or reliability problems and require solutions to be managed by the current proposed protocol.

### 9.2.1 Inter-domain link failure in an Ingress Tunnel Router (ITR)

Next, we will discuss and describe the actions that are executed in order to solve an inter-domain link failure in the presence of LRP. In step 1 of Figure 9.2, the traffic is sent to the edge router (ITR1), which is responsible for encapsulating the traffic and send it through international links to the destination. When the international link fails (step 2) the ITR1 automatically detects this event and in real time forwards all the incoming traffic to the other ITR belonging to its LRP group (step 3). ITR2 has the correspondent mapping since it shares the LRP Group with ITR1 and now is in charge of encapsulating and sending this traffic to its destination (step 4). On the other hand, by means of the internal routing protocol (running in the AS), the failure of the international link is notified to update the routing tables and hence the traffic is rerouted (step 5). In time, the LISP Control Box (LCB) would be responsible for reconfiguring the mapping of the different ITRs with the purpose of load balancing the outbound traffic (step 6). Finally, the traffic is rerouted according to internal routing policy (step 7). In short, in this scenario our border architecture prevents packet loss and in particular the sending of data-probes.

### 9.2.2 Ingress Tunnel Router (ITR) failure or unexpected shutdown

In the case that an Ingress Tunnel Router (ITR) fails or is shutdown, gigabits of data would be lost. To overcome this issue, the LISP Control Plane must converge to the Interior Gateway Protocol (IGP) running in the AS, in order to send back first gigabits of data-probes and after successful mapping, send data according to the normal procedure. The following describes the steps followed by the LRP to prevent any loss of data and to avoid sending data-probes. In step 1 in Fig. 9.3, the traffic is sent to the edge router (ITR1), which is responsible for encapsulating the traffic and send it through the international link to its destination. When ITR1 fails (step 2) the HSRP that runs between ITR2 and ITR1 converges, and automatically ITR2 assumes the role of Master of the LRP group and forwards all traffic that arrives (step 3) in real time. The convergence of HSRP is much faster than any IGP, such as Open Shortest Past First -

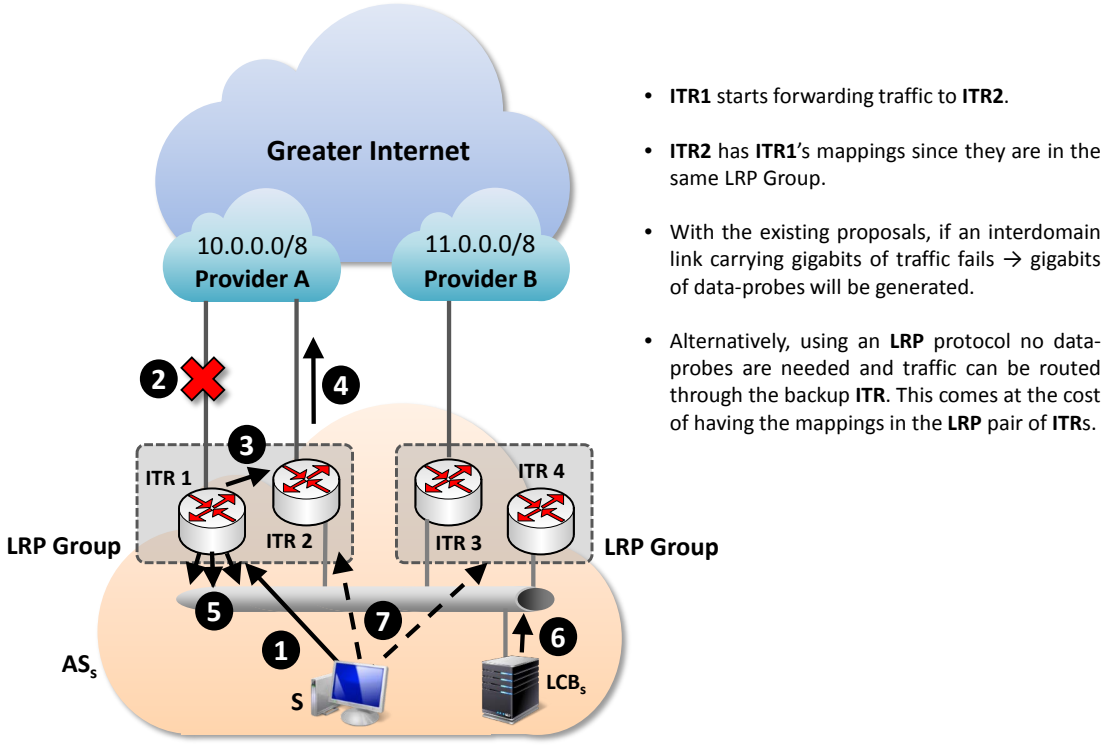


FIGURE 9.2: LRP: Inter-domain link failure.

OSPF. In turn, the internal routing protocol that is running notifies the failure of ITR1 (step 4) to update the routing tables and hence allowing the traffic to be rerouted. On the other hand, the LISP Control Box (LCB) would be responsible for reconfiguring the mapping of the different ITRs to balance the outbound traffic load (step 5). Finally, the traffic is rerouted according to the IGP (step 6). In short, in this scenario our border architecture minimizes the packet loss and in particular the sending of data-probes.

### 9.2.3 Ingress Tunnel Router (ITR) Mapping Miss

In this scenario, an action of Traffic Engineering or an internal failure (step 1 in Figure 9.4) makes the traffic to be rerouted. When the packet reaches a border router (step 2) that has no corresponding mapping, this router makes a broadcast to other LRP Groups of the packets that are arriving (step 3), and in turn, the LRP Group sends a Map-Request to the LCB (step 4) that is responsible for handling all mappings within an AS. A LCB (LISP Control Box) is an entity responsible of all mappings within an AS which might be a standalone device or run as an instance of a PCE. The LRP Group that owns the required mapping, sends it via unicast to the LRP Group responsible for these packages (requester) (step 5), and encapsulates and forwards the traffic. While traffic is derived from the LRP Group mapping holder (step 6), the LCB sends to the LRP Group who sent the Map-Request the mapping necessary to encapsulate the packages

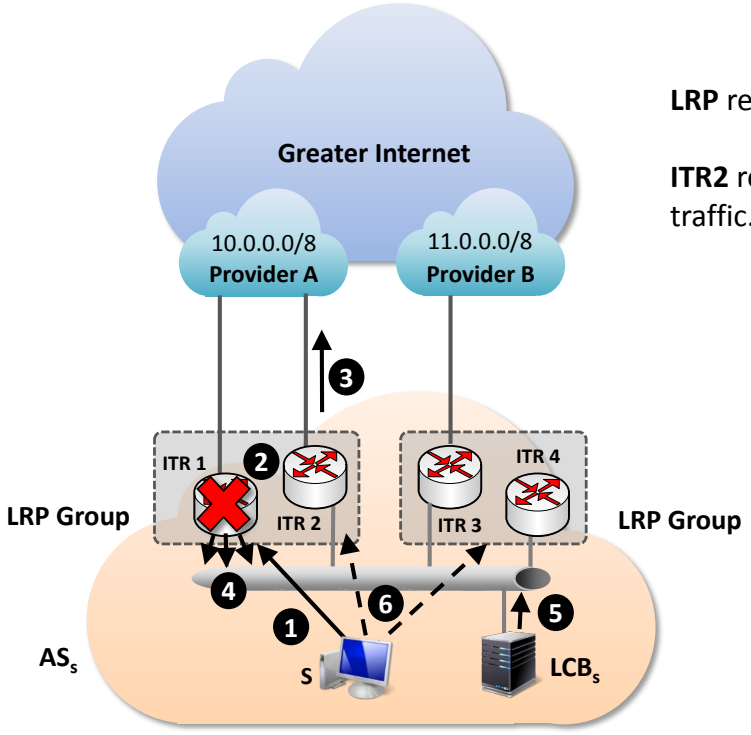


FIGURE 9.3: ITR Failure.

(step 7). Finally, the LRP Group can now encapsulate packets and, therefore, makes the package forwarding through the LISP data plane (step 8). In conclusion, in this scenario our border architecture prevents packet loss and in particular the sending of data-probes.

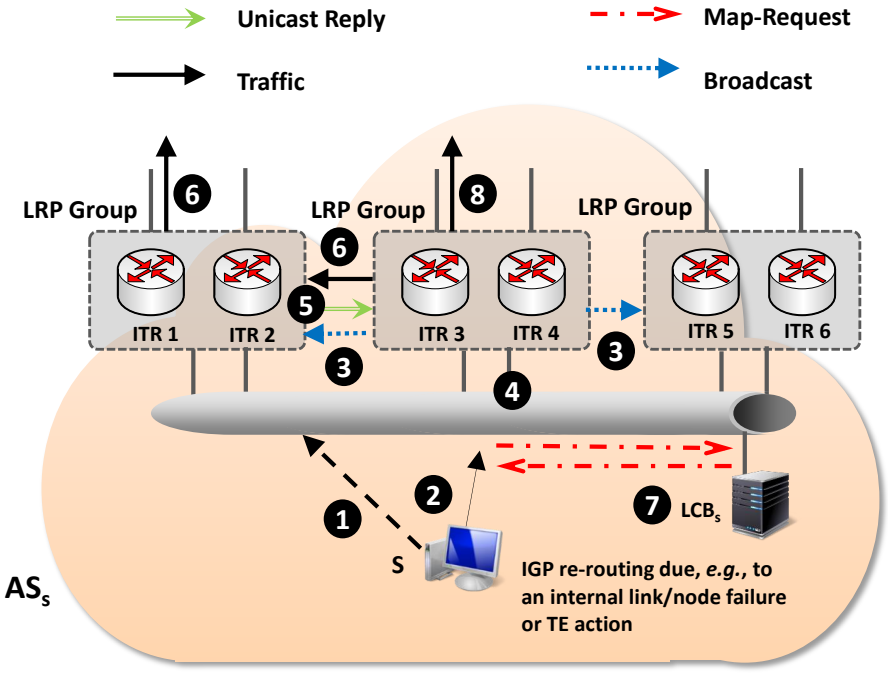


FIGURE 9.4: ITR has no mapping from EID-to-RLOC.

# Chapter 10

# The Future Internet

*In this Chapter we describe the fundamentals of TARIFA, an approach for next generation networks and finally, introduce a novel addressing scheme for a service-oriented architecture.*

Network communications have rapidly evolved in the past years, this growth has led to numerous challenges, many of which remain unsolved or in other cases when solved, have derived in new issues, due mainly to inflexibility of the initial hierarchical layered model of current Internet architecture. In the past years, emerging of new applications, services, users growth, mobility requirements, among others, have become issues of high interest under the scope of many researchers on the Internet area. As a result of these efforts, research projects such as GENI [26], 4WARD [27], DONA [28], TRILOGY [29], or PSIRP [30] have been developed in order to discuss novel ideas and present novel network architectures. TARIFA [31] also figures as an initiative towards developing a clean slate Future Internet (FI) architecture, where the network paradigm is no longer based on the interconnection of machines/interfaces but on the interconnection of services/resources, an approach that seems to meet much better both present and future users requirements. TARIFA takes as basis a service-oriented paradigm for service provisioning in Future Internet, where communications can be composed on the fly and dynamically adapted by exploiting reusable components or services.

A major concern, however, for making possible this shift of paradigm is the addressing scheme used in the Internet. In the past years, scalable routing and addressing architectures for the current Internet have become target of many research efforts. Concerns related to the scalability of the routing system and the impending exhaustion of the IPv4 address space have led to important proposals based on the "locator/identifier separation" [141].

Multi-homing and traffic engineering represent two of the main factors contributing on routing tables exponential growth, due basically to IP addresses semantic overload. Current Internet combines two functions: location (point of attachment to the network, where) and identification (who), in other words, current IP addresses have dual semantics, identifiers and locators are under the same numeric space. Several approaches have been pursuit in order to solve these open technical issues, SHIM6, HIP, LISP are just a few to name.

Several reports can be found in the literature claiming that the semantic overloading of IP addresses is hindering the progress toward the deployment of service-centric architectures [4]. In the context of Internet mobility this has serious and well-known scalability issues, hence avoiding this semantic overloading is a key objective while designing new addressing schemes. Taking these facts as basis, herein we present a novel addressing scheme which aims to overcome the addressed issues to yield several advantages, including improved scalability for the routing system, by dealing with the Internet routing table exponential growth. This scheme has been designed and evaluated as part of the TARIFA project [31].

## 10.1 A Service-Centric Internet Architecture: TARIFA

TARIFA [31] aims at defining a clean slate approach to a Future Internet architectural redesign, based on a role-based paradigm consisting of non-divisible, or atomic, functions. TARIFA architecture is service-oriented, enabling dynamic composition of services and its adaptation, taking into account context status and its variations. Thus, services become the fundamental functional components to be composed, taking into account the specific requirements of each communication and allowing to adapt them to fulfill context variations. Service-oriented approaches are foreseen to be suitable for providing seamless communications which allow to deal with heterogeneity and dynamic conditions of the network.

The TARIFA approach proposes that network maintains context information by means of distributing it among network nodes. We take advantage of the distributed nature of this approach in order to achieve a high level of scalability and flexibility. Thus, context-awareness is enabled by making context information available to nodes in the Network.

The TARIFA approach is service-oriented, as it allows to discover, combine and dynamically adapt the functional service blocks, according to context variations. Under this vision, we worked towards the proposal of a new addressing scheme oriented to the real

requirements of a service-centric architecture. Next, we will introduce the main ideas of our proposal, followed by an analytical validation of the concepts herein introduced.

## 10.2 A Service-Centric Addressing Scheme

### 10.2.1 Overview

In the context of the newly proposed Network architecture a new addressing scheme is proposed in order to suit the needs of reachability and overcome current issues related to routing and semantic overload of IP addresses. Our proposal relies on the separation of the space of locators and identifiers, due to the correspondence of this model to a service-oriented approach, where mobility and service migration have quite high probability.

Under this model, the generation of two clearly separated spaces (naming and location), allows us to introduce two new important concepts, namely, the TARIFA Identifier (from now on referred to as TID) which permits the unique identification of services and resources, regardless its actual location and the TARIFA location (from now on referred to as TLOC) which indicates the current location of the identified resource or service over the network. The main features of the proposed addressing scheme as listed as follow:

### 10.2.2 Naming Space

As introduced previously, the naming space refers to the space of identification of services and resources available in the network. Under our model, TIDs are fixed length structures following a given hierarchy, which allows the unique global identification of services and/or resources. Our addressing scheme, does not assume any kind of restrictions over what can be identified, in other words, tangible (physical, e.g. PC, PDA, etc.) or intangible (virtual, e.g. files, services or applications) may have their own identifier.

The generic format of a TID is shown in Figure 10.1. From the figure it is possible to distinguish three fixed size fields, namely:

- *Entity*, this 64 bit field refers to the root entity instantiating a given service or re-source. As in cur-rent citizen identification systems, where each citizen is identified through a unique number, our approach intends to assign a domain of identification to each citizen, organization, virtual group or enterprise, under which each entity will identify the set or subset of services and resources of its interest.

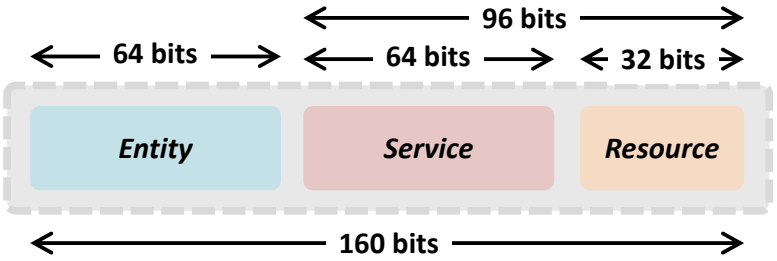


FIGURE 10.1: TID format and size for Service-Centric architecture.

- *Service*, this 64 bit field allows to address all available services for a given entity, by this, one entity is capable of addressing a total of 264 services, which is actually the square of the current internet.
- *Resource*, this 32 bit field allows to address all available resources for a given entity.

A key feature of the proposed identification scheme is its two-fold purpose, on one side it serves to uniquely identify services or resources, and on the other side it provides com-posed identification, meaning that we can identify services related to a concrete resource (e.g. a printing service dependent of an specific printer).

A TID is a 160 bit fixed length structure. Our proposal pushes towards fixed length identifiers as it has been proven to be the best alternative for hardware scalability, simplicity and cost-reduction. Current network hardware is especially designed to optimize forwarding tasks, for which fixed length is critical. On the other hand, transport networks are moving towards optical solutions, in this matter, variable length identifiers would imply major electronic processing which goes against the all-optical trend.

Determining the fixed size of the proposed TID is an important challenge, as it must suit the complete space of identifiable objects. Based on the previous idea of providing a block of identifying space for entities to uniquely name services and resources under their own space, we must face a new challenge related to the number of entities that our space of identifiers shall be able to address. The answer boils down to overestimating the space to be addressed. To this end, we assume that today's worldwide current population is around 7 x109, this implies that 33 bits would be required to address the complete world's population. According to the United Nations Demographic Forecast [10] the population for 2300 is estimated to be around 8.97 billion, meaning that 34 bits would be enough to globe this figure. Taking in consideration that many other entities (enterprises, organizations, logical groups, governments) could be provided of a block of identifiers, a gap of 30 extra bits seems to be a suitable number. In this sense, 264 entities are addressable over a TARIFA Network. Then, each entity is able to address 296 objects, semantically differentiated in two fields: service (64 bits) and resource (32

bits). The main advantage of such a type of identification is that it allows automatic identification clustering, or aggregation. An important aspect to highlight from the proposed solution, is to make clear that for networks with no infrastructure and sensor networks composed of low capacity devices, these may internally be managed by the manufacturer assigned identifier, which would not be visible to the global network, but managed through an aggregation point ruled under the TARIFA identifying scheme which is actually visible out of its boundaries. In this sense, both schemes can coexist.

TID's express the binary or numeric representation of identifiers for services and resources over a TARIFA Network. But what actually seems as important, is to provide users with the capability of consuming already well-known services, and for this, knowledge on the desired service identifier is required. Since numeric representations and especially long formats do not provide friendly-human readable identifiers, we propose the existence of friendly names or **TARIFA F**riendly **N**ame **I**dentifier (*TFNI*). A *TFNI* will provide users with the capability of reminding identifiers to well-known services and direct access or request of them. A *TFNI* will typically be a string-based identifier, following the same hierarchical structure proposed for TIDs, that is a triplet entity-service-resource. This means that all services under the same entity will share a common prefix, by them all being services of the same entity (e.g. companyA.videoServ, companyA.printServ, etc.). This new concept within the Project faces a new challenge, this is, there must exist a supporting system capable of solving *TFNI* into TIDs, this is the TARIFA Resolution Name System (TRNS) a concern left for future work.

### 10.2.3 Location Space

The clear separation of the addressing space, allows the generation of a location space, referred to the position of identified services or resources in the Network. A TARIFA Locator (TLOC) is a fixed length structure reflecting the position of a resource or service in a binary/numeric format.

Given the basis of the TARIFA Project [31], this is, a service-oriented approach, locators do not aim to be only pointers to physical objects, but to physical and virtual entities as well. This raises a new challenge related to mapping from identifiers to locators for virtual entities, this issue will be discussed further in the next section. As TIDs, the proposed TLOCs are globally unique, meaning they have representation over the entire scope of the Network. Our proposed model, provides flexibility regarding the location scheme, allowing the coexistence of different representations such as, flat scheme, hierarchical scheme or those based on geolocation, among many others. A fixed number of bits (8 bits) is reserved for indicating the location scheme. It is important to highlight

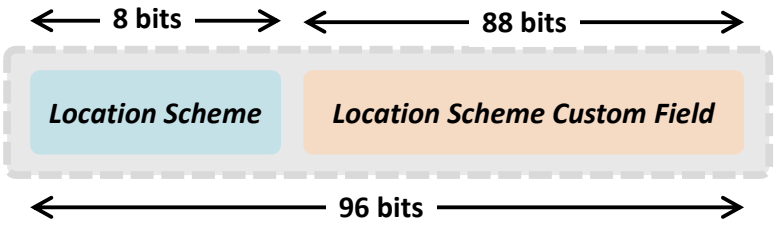


FIGURE 10.2: TLOC format and size for Service-Centric architecture.

that a subset of these 8 bits remain available for future options, such as security policies, QoS, etc. The remaining 88 bits are used for direct representation of the locator for the underlying scheme.

In order to assure that a geolocation scheme under the proposed model will provide the precision and resolution required for a feasible addressing scheme, we proved that an 88 bit locator field provides high quality resolution. The total area of the planet's surface is around $S_{earth} = 510.086 * 109\ \ m^2$. For a desired resolution of 1 mm2 (this means being able to locate objects as small as $1mm^2$ above the $S_{earth}$) 69 bits are needed, this is less than the 88 bits free for location assignment. With this number of bits a resolution in the order of $nm^2$ can be achieved, this is far away from a minimum feasible resolution.

### 10.2.4 ID/LOC Mapping

With the separation of identifiers and locators in two different spaces, a new concern emerges, this refers to the need of a mechanism able to map identifiers to locators. Many approaches can be found in the recent literature dealing with such a mapping system [22][23][200]. The proposed identifier-to-locator mechanism within the context of a service-oriented network is based on DHTs. For this purpose, all Manager Elements in a TARIFA-compliant Network are part of a mapping overlay. These nodes form a Chord ring topology and are responsible of a group of (key, values) pairs, where keys are identifiers and values are locators. There are two main advantages in mapping to a DHT-based implementation. One is that this mapping system is fully decentralized. A second advantage is scalability (as demonstrated in [31]), this scheme experiences sub-linear growth with the number of addressable nodes n, namely, log(n). The proposed mechanism was implemented in the framework of the TARIFA project in the form of a mapping overlay. An important aspect related to the mapping system in service-oriented network architectures, is the mapping of TIDs that identify virtual entities, as in current Internet this is not an issue. As such, we consider two potential lines of work for handling the previous concern:

- By indirect mapping, which means that by looking up for a virtual entity identifier in the DHT, the mapping would be done to the physical object's identifier that currently contains the identified object, and a recursive lookup would follow up.
- By direct mapping, this means that the DHTs are now capable of relating TLOCs to virtual identifiers, supporting updates directly in the location of services or data.

### 10.2.5 Experiments and Performance Evaluation

In order to evaluate the performance of the proposed ad-dressing scheme, we developed an analytical methodology aimed to estimate the reachability and limitations of the identification space. According to the United Nations Demographic Forecast [201] the population 289 years from now (2300) is estimated to be around 8.97 billion, meaning that for the proposed space of identification a single entity, represented by a person, would be able to address a total of 1.62E+38 services/resources, or otherwise seen, there would be 2.87E+29 services per square centimeter, minimizing the possibilities of exhaustion of the identification space. An even more challenging issue arises while foreseen the scalability of the proposed mapping system, to actually support the increasing number of services to be published.

This section presents an analytical evaluation of the pro-posed mapping system, in order to estimate the size and characteristics that define it, as a function of the chord ring dimension and storage and forwarding rate capabilities of each node. This analysis will give us a clear estimation and definition of mapping parameters (i.e., number of elements, storage properties, etc.) that will give support to the complete space of identifiers and their mappings.

The Chord ring is composed of all the set of nodes within the mapping overlay, and from now on we will refer to these nodes as Chord Nodes [202]. For simplicity purposes, some assumptions are made in advance, we will assume that services are reachable only though one locator (TLOC), although in fact it can be reached through several locators.

The following analysis aims to determine the size of the space of identifiers, as a relation of the number of elements in the mapping overlay and its properties (i.e., storage and for-warding rates) in order to evaluate scalability within current available technology, so as to support scaling according to the expected growth.

The number of mapping entries is represented by the total number of TIDs available in the system (denoted by TTIDs). According to the proposed scheme, TTIDs may take a maximum value of 1,4615E+48 ($= 2^{160}$), in the case that the complete space of

identifiers has been assigned. This number is ideally an upper bound that far exceeds the total space of identification for services over the Network. Due to previous assumptions, a mapping entry is a pair TID-TLOC with a total length of 32 bytes (see expression 10.1), from now on, we will denote the length of a single entry of the mapping system as $\mathcal{L}_{entry}$.

$$\mathcal{L}_{entry} = 160bits + 96bits = 256bits = 32bytes \tag{10.1}$$

Next, we present the analytical expression for the total number of Chord Nodes (denoted by $T_{chord\_nodes}$) as a function of the total number of identifiers ($\mathcal{T}_{TIDs}$) and storage capacity ($S_{chord\_node}$) of a single node. We assume that mapping entries are uniformly distributed over the mapping system.

$$T_{chord\_nodes} = \frac{32bytes \cdot \mathcal{T}_{TIDs}}{\mathcal{S}_{chord\_nodes}} \tag{10.2}$$

From the previous expression, we can notice that the total space of addressable identifiers ($0 < \mathcal{T}_{TIDs} \leq 2^{160}$) is a trade-off among the total number of elements conforming the mapping overlay and the storage capabilities of each of these nodes. It is worth highlighting that for evaluation simplicity, Chord Nodes are supposed homogeneous, this means that all Chord Nodes have equal performance and attribute values; for our consideration all management nodes have the same storage and forwarding capabilities.

Besides nodes storage capacity, forwarding rate is another metric to take into account for further scalability evaluation. Let's denote by Rchordnode the number of requests per second that a Chord Node is able to handle. Chord Nodes requests are represented by both, update and lookup messages. Update messages aim to introduce new mappings in the system or update current entries due to mobility, for example. Whilst, lookup messages aim at requesting the retrieval of a mapping. Mappings are requested by means of an specific service identifier (TID).

Let $P_{requests}$ be the percentage of requests per second, and let $P_{requests}.\ \mathcal{T}_{TIDs}$ be the total number of mapping requests per second in the system. The values used for $P_{requests}$ are taken from conducted estimations made in **[HON09]** under practical environment for Internet traffic. Besides, from previous research efforts [202] the average routing path length for a DHT-based Chord Mapping System is:

$$\langle avrg\_routing\_path \rangle \geq \frac{1}{2} \cdot log(T_{chord\_nodes}) \tag{10.3}$$

Let's assume that requests are evenly distributed among all Chord Nodes in the mapping system. Taking all these assumptions into account, we can express the forwarding rate of each Chord Node ($\mathcal{R}_{chord\_node}$) as a function of the average routing path length and total number of requests per chord node, this is:

$$\mathcal{R}_{chord\_nodes} > \frac{1}{2} \cdot Log(T_{chord\_nodes}) \cdot \frac{P_{requests} \cdot \mathcal{T}_{TIDs}}{T_{chord_nodes}} \tag{10.4}$$

From the above equation, we can denote the total space of identifiers ($\mathcal{T}_{TIDs}$) as:

$$\mathcal{T}_{TIDs} > \frac{2 \cdot \mathcal{R}_{chord\_node} \cdot T_{chord\_nodes}}{Log(T_{chord\_nodes} \cdot \mathcal{P}_{requests})} \tag{10.5}$$

The first step into the scalability evaluation is determining the growth of the mapping space over storage and forwarding rates. By means of current technology, the presented analysis can be seen as a lower bound for the addressable space of identifiers. It is clear that technology is pushing towards higher storage capacities and maximum forwarding rates, in this sense, the addressable space of identifiers will tend to increase with major node capabilities.

The scalability evaluation is done for two scenarios, according to different forwarding rates for current technology. For the first scenario of evaluation, as in [22], we assume the nodes forwarding rate (Rchordnode) to be $10^8$ packets per second. Figure 10.3 is a three-dimensional representation of the size of the space of identifiers, as a relation of the percentage of requests into the mapping system and the total number of mapping elements (Chord Nodes). As shown in Figure 10.3, the addressable space of identifiers increases for low percentage of mapping requests and high deployment of Chord nodes, this result is within the expected behavior, as the number of identifiers that our system can support, is expected to be higher as the incoming requests decrease and there are more elements with homogeneous capabilities of giving support to the total space of identifiers. The maximum number of locators (TLOCs) to be addressed over a TARIFA environment goes up to 3.09 x$10^{26}$ (=$2^{88}$, where 88 is the total number of bits designated for location). As shown in Fig. 10.3, with a structure of mapping elements of barely 3.1% of all addressable locations, a total of 7.74 x $10^{35}$ identifiers can be mapped into the system, which is 21 orders of magnitude greater than the base numbers found in [31] and 26 orders of magnitude greater than the addressable space reached with IPv4.

For the second scenario of evaluation, we take as reference the forwarding rate ($\mathcal{R}_{chord\_node}$) of Junipers T Series Core Routers, being this 30x$10^{12}$ packets per second. As in the previous analysis, Fig. 10.4 depicts the growth of the space of identifiers as a relation of

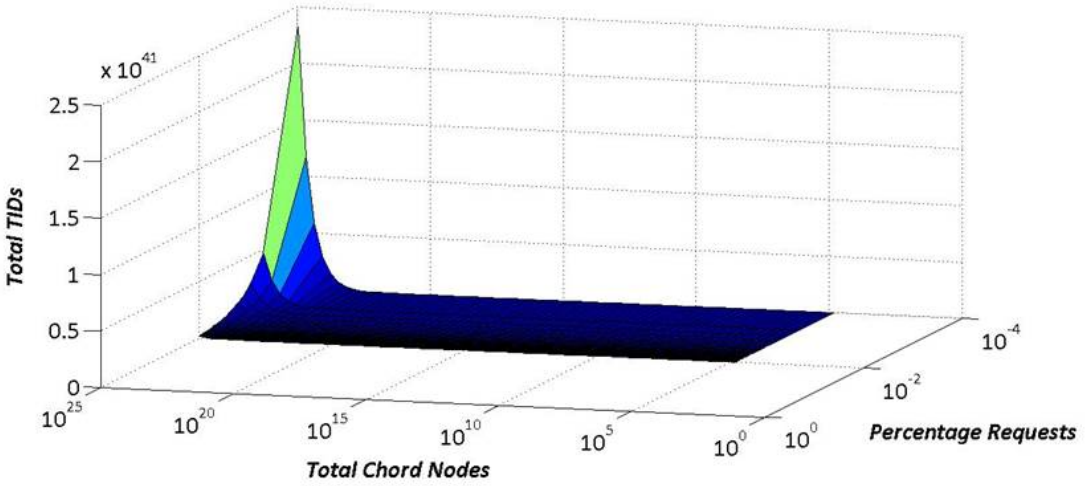


FIGURE 10.3: Mapping Space Dimension as a function of percentage of requests and total number of chord nodes in the system for Rchordnode = 10 $^{8}$ [pps].

the percentage of mapping requests and total number of Chord Nodes. The idea behind depicting the behavior of the total addressing space for the previous described scenarios, is to evaluate tendency, aimed to prove scalability within more challenging capacities. From the plot in Figure 10.4 we can conclude that for a fixed forwarding rate of 30x10$^{12}$ [pps] even under an scenario with less advantageous conditions, this is, highest percentage of mapping requests to be resolved and lowest number of nodes managing the mapping system, the space of identifiers is 1010 times the number of persons on earth. On the other hand, for the same conditions evaluated for the previous scenario, with a forwarding rate increased to the order of Tera [pps] the addressable space increases in 6 orders of magnitude.

The dimension of the addressed space of identifiers directly affects the storage requirements per Chord Node. Figure 10.5 depicts the storage requirement per chord node for the first studied scenario (forwarding rate in the order of 108 [pps]), for different percentage of incoming requests and variations over the total number of elements in the mapping system. As seen, storage requirements are around the order of terabytes, a result that is in line with current available technology, meaning that, chronological evolution of storage technology will positively influence the increase in size of the address space.

The results obtained from the previous analysis can actually be enhanced by means of two mechanisms: caching and aggregation. Caching would decrease the number of incoming mapping requests into the system, while aggregation would reduce the storage space required for keeping mapping entries within Chord Nodes mapping tables. These

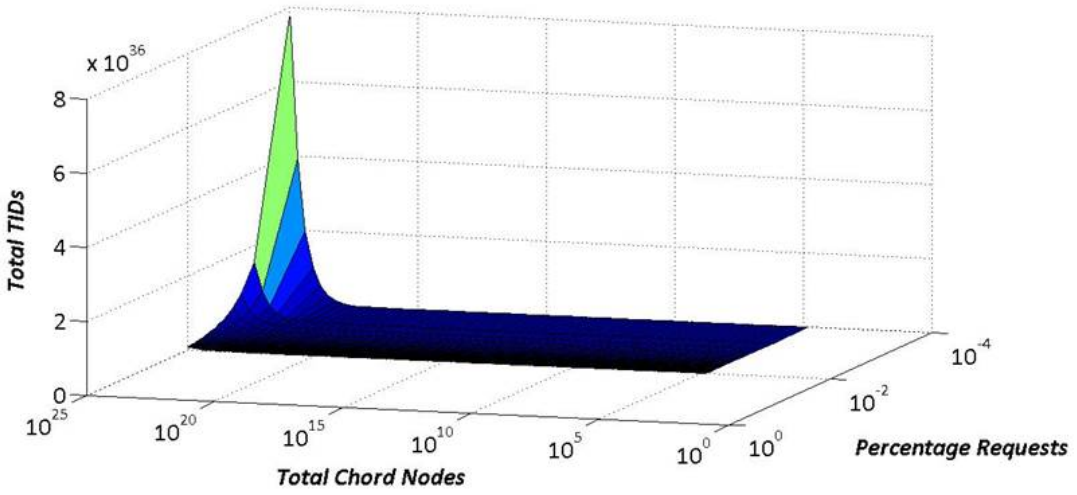


FIGURE 10.4: Mapping Space Dimension as a function of percentage of requests and total number of chord nodes in the system for Rchordnode = 30.10 $^{12}$ [pps].

mechanisms are out of scope of the current stage of this analysis, but will be considered on future work.

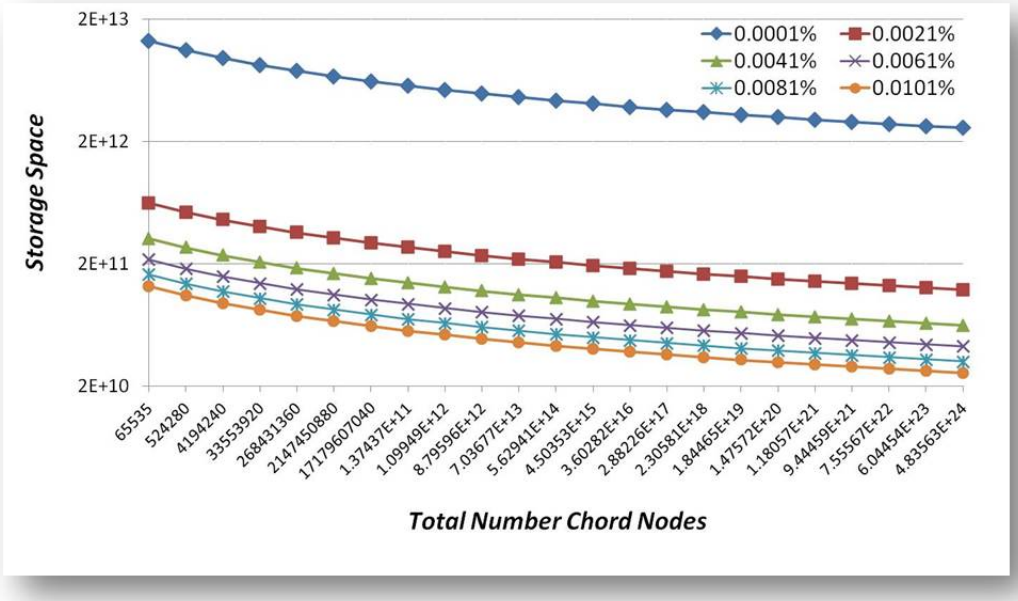


FIGURE 10.5: Storage Space Requirements as a function of the total number of chord nodes in the system for different incoming requests, for first scenario Rchordnode = 10 $^{8}$ [pps].

# Part V

# CONCLUSIONS AND FUTURE WORK

# Chapter 11

# Conclusions

This thesis has highlighted relevant semantic issues in the field of networking. Due to its broad scope, we have focused on two well-known subfields, namely, network configuration management and addressing architectures. Overall, the motivation to analyze the semantic issues in these areas is that problems date from the earliest days of networking, and despite numerous efforts from both, academia and industry, they continue in force in the current Internet—up to the point that, the most primitive technologies are still the preferred mechanisms in place. Given that, semantic issues in both fields are dissimilar in nature we have analyzed these problems separately.

Based on the analysis led in the scope of network configuration management, we have shown that despite the most recent emergence of technologies such as, NETCONF, SDN supported through OpenFlow and OFCONFIG, as well as the consolidation of Web Services, Command Line Interfaces (CLI) do not only continue to be the leading mechanisms for the configuration of legacy infrastructure, but even more, in the presence of next-generation technologies they are required as a means to complement configuration capabilities—whenever heterogeneity or data models hinder the remote configuration of network devices, this is particularly the case of hybrid SDN networks. Furthermore, as CLIs are essential resources for the exchange of configuration data with network human operators and thus, are largely based on natural language text, it seems reasonable to adapt forms of natural language analysis of the Artificial Intelligence field to enable semantic abstraction. Based on this, the challenge of enabling semantic interoperability across heterogeneous environments has been addressed by developing a novel Ontology-Based Information Extraction (OBIE) system from the CLI of network devices to ease the complex task of device configuration. We have shown the potential of Information Extraction (IE) techniques and other semantic technologies to automatically exploit the knowledge natively provided by vendors in the form of natural language in their

CLIs. Furthermore, we have developed an ontology for the network device configuration domain (ONDC) which provides a semantic backbone for the OBIE System herein developed. The ONDC ontology is further used to *(i)* guide the IE process from the CLI and *(ii)* enhance our assessment by exploiting the ontological structure. We have also developed an extraction method which relies on shallow NLP tools, lexical matching, clustering techniques, ontology reasoning and relatedness computation to perform automatic instantiation of CLI commands and variables into their semantic categories.

Based on the performance evaluation we have shown that the use of ontologies in conjunction with other semantic technologies for IE offer a promising path to mitigate the efforts of network device configuration. We have also concluded on the suitability of our design decisions for which we achieved a maximum performance of 91% precision and recall. Moreover, we have motivated the use of our OBIE system in the context of current multi-layer core infrastructures as a means to support the configuration capabilities of solutions aimed to overcome the isolation between management ecosystems. For instance, the case of the ONE Adapter, for which we have tested our model and validated the ability to resolve semantic and syntactic adaptations for a set of use cases of high impact for network operators, including, IP link provisioning, OSPF routing protocol setup, MPLS configuration and IP offloading in optical networks.

Finally, we have discussed on the major limitations of the current Internet addressing architecture and analyzed the impact over network performance and other network functions. Given the fact that current IP address semantic overload largely contributes to the problems related to mobility, traffic engineering, resilience and multihoming, we have examined two potential paths of solutions. The first, an evolutionary path achieved through the adoption of the LISP framework—a paradigm that implements new semantics for IP addresses. The second, a disruptive path wherein a novel addressing scheme is proposed for a service-centric network architecture. For the first line of research, we have shown the limitations surrounding one of the most recent proposals for Control Plane technologies for LISP and proposed the concepts for a new protocol, namely, the LISP Redundancy Protocol (LRP) aimed to overcome the scalability issues and improve reachability and reliability for an Autonomous System. For the second line of research, we proposed a new semantic-based addressing scheme for a service-oriented Future Internet network architecture. We have introduced separate spaces for host identification and location and conducted an analytic evaluation. Our evaluation results show that the proposed scheme meets future Internet requirements, including, suitability in highly demanding addressing scenarios.

In the next chapter, we highlight the author's view on potential directions for extending the reach of the work developed in this thesis.

# Chapter 12

# Future Work

*In this chapter we outline remaining open issues derived from the work developed in this thesis and draw potential future lines of work which might be considered by other researchers.*

## 12.1 Semantic-Based Network Configuration Management

Overall, future lines of work can still be explored from an algorithmic perspective, which include, *(i)* enhancement of the learning algorithm based on historical instantiations, *(ii)* enhancement of the IE process from web-based resources; *(iii)* WordNet verb recognition support; *(iv)* weigh of CLI data based on the structural source of the information; and finally—based on the notion that good suggestions are already produced by our system but still holding for final verification—*(v)* integration of a human validation stage in the final loop to improve the instantiation process to its maximum. Next, we will provide some further insights on these potential lines of work.

**Historical Disambiguation.** The hierarchical structure of CLI environments is key to our learning algorithm. We have exploited the information implicitly embedded in the CLIs hierarchy by computing semantic relatedness among contiguous levels—within a configuration statement—in order to increase accuracy of the instantiation process. However, we believe we can further exploit the semantics of the hierarchy by taking into account previous instantiation decisions, i.e., considering the decisions made for previous configuration statements. This would provide an additional scope to the decision maker, to disambiguate the sense of a term based on previous decisions. Indeed, this comes at the cost of keeping state, an issue that needs to be further studied. Consider for instance, the following scenario. If the atomic operation ⟨Set Interface MAC_Address⟩

was identified for the command ⟨mac⟩ for the previous configuration statement, with high probability the ⟨address⟩ command in the subsequent statement (for the same level) refers to the atomic operation ⟨Set Interface IP_Address⟩ rather than ⟨Set Interface MAC_Address⟩.

**Enhance IE from Web-based resources.** We suggest the use of the Web as a big corpus to extract additional information that can improve the instantiation algorithm. This information can actually be used not only to aid in the process of ontology population, but also to enhance the meta-ontology while new capabilities or technologies appear, which is actually concerned with the field of ontology learning.

**Include WordNet in verb recognition.** In the context of our algorithm, the identification of verbs from the CLI is key to determine the category of an action, and thus discern over the type of operation. In case that a verb does not map to any category of our meta-ontology, we can rely on a lexical database, such as *WordNet*, for synonym lookup and further determine the most similar semantic category.

**Weigh CLI Data.** As previously stated—in an effort to guide network administrators through the configuration environment—*help descriptors* tend to narrow down the semantics of *commands* and *variables* to a set of common and shared conceptualizations of the domain. While *commands* and *variables* are typically constrained to single keywords which are at the same time less expressive and subject to vendors' custom terminologies. In such a context, one could think rational to give higher weigh to decisions made over information extracted from *help descriptors* rather than *commands* and *variables*. Our current algorithm is indifferent to the source of knowledge.

Moreover, other lines of work related to potential applications of our system can be further explored, namely, *(i)* Semantic-Web integration and *(ii)* extensibility to other domains.

**Semantic-Web Integration.** Integrating our application into the Semantic Web would provide the ability to generate semantic contents of the network device configuration domain for the Web.

**Extensibility to other domains.** As successfully applied to the field of networking, it would be of great interest to prove our systems extensibility to other domains wherein configuration is also based on Command Line Interfaces.

## 12.2 Semantic-Based Approaches for Network Addressing

Regarding the proposed LISP Redundancy Protocol (LRP), our network scenario assumes that LRP nodes are all part of a broadcast domain. In current ISPs core networks this is not always the case, with the introduction of new technologies like MPLS—where edge routers reach each other using LSP (Label switched Path) across an MPLS core. In such a network scenario a single broadcast domain does not exist anymore. A similar problem appears when BGP confederations are introduced. To overcome these challenges, extensions to LRP must be done and configurations in edge routers have to be carefully taken into account. Furthermore an implementation of the LRP Protocol is required to obtain measurements on the convergence time in the network, considered as a primary metric for determining the scalability and efficiency of routing schemes.

# Appendix A

# European Projects

Some of the contributions of this thesis have been partially used in the context of the following projects.

***European Projects***

- **FP7 Project ONE** [111]: Towards Automated Interactions between the Internet and the Carrier-Grade Network Management Ecosystems. **FP7-ICT-2009-5** (2010-2014)
- **FP7 Project TEFIS (Opener)**: TEstbed for Future Internet Services, TEFIS Project **FP7-258142**, (2010-2013). As part of the TEFIS Project: OPENER, the Open and Programmable ENvironment for Experimenting with Routers.

***Integrated Research Project (i2cat)***

- **TARIFA** [31]: The Atomic Redesign of the Internet Future Architecture. **i2cat** (2010-2011).

## A.1 ONE in a Nutshell

***Towards Automated Interactions between the Internet and Carrier-Grade Management Ecosystems***

Although IP and transport networks are typically deployed in tandem, their intrinsic differences have profoundly segmented the way in which operators manage these infrastructures. Most operators have equipment from at least two vendors at each layer.

Clearly interoperability at the data and control planes is a must when purchasing equipment from different vendors, but when it comes to the management plane, the isolation is unavoidable, in practice. The lack of commercial solutions capable of providing automated coordinations of management procedures, such as the orchestration of business practices between layers and vendors, poses considerable challenges and overheads to operators both in terms of CAPEX and OPEX.

***What does "ONE" do?***

- **Coordination without integration:** multi-vendor and multi-layer NMS coordination.
- **Semantic Adaptation:** vendor-independent configurations not only in the IP layer but also multi-layer (optical).
- **Programmable Network Management:** "Manual where needed and automated where allowed." Smart analytics to create, manage, and redefine network management procedures via workflows.
- **Native Third-Party Support:** Path Computation Element (PCE), OpenFlow interfaces in the optical layer, AAA, etc.

Figure A.1 depicts a general diagram of the functionality of the ONE middle-box in a multi-layer and multi-vendor scenario.

***What does "ONE" not do?***

- **Yet-another NMS:** Lightweight system adapter independent of any NMS.
- **Yet-another control plane client:** but is control-plane friendly and can "talk" to any control-plane implementation.
- **Yet-another virtualization tool:** it offers simple adaptation of semantics of whatever vendor is deployed.

## A.2 TARIFA in a Nutshell

TARIFA [31] aims at defining a clean slate approach to a Future Internet architectural redesign, based on a role-based paradigm consisting of non-divisible, or atomic, functions. The TARIFA architecture is service-oriented, enabling dynamic composition of

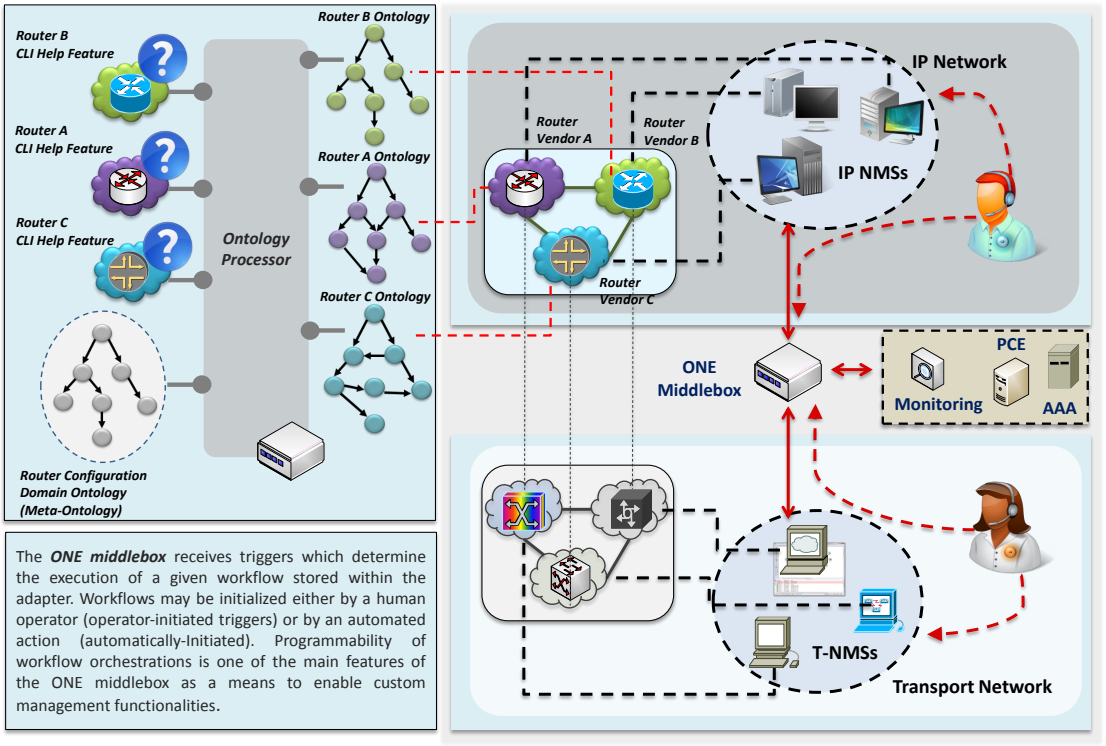


FIGURE A.1: Functionality of the ONE middle-box in a Multi-Layer and Multi-Vendor Scenario.

services and its adaptation, taking into account context status and its variations. Thus, services become the fundamental functional components to be composed, taking into account the specific requirements of each communication and allowing to adapt them to fulfill context variations. Service-oriented approaches are foreseen to be suitable for providing seamless communications which allow to deal with heterogeneity and dynamic conditions of the network.

The TARIFA approach proposes that network maintains context information by means of distributing it among network nodes. We take advantage of the distributed nature of this approach in order to achieve a high level of scalability and flexibility. Thus, context-awareness is enabled by making context information available to nodes in the Network.

The TARIFA approach is service-oriented, as it allows to discover, combine and dynamically adapt the functional service blocks, according to context variations. Under this vision, we worked towards the proposal of a new addressing scheme oriented to the real requirements of a service-centric architecture.

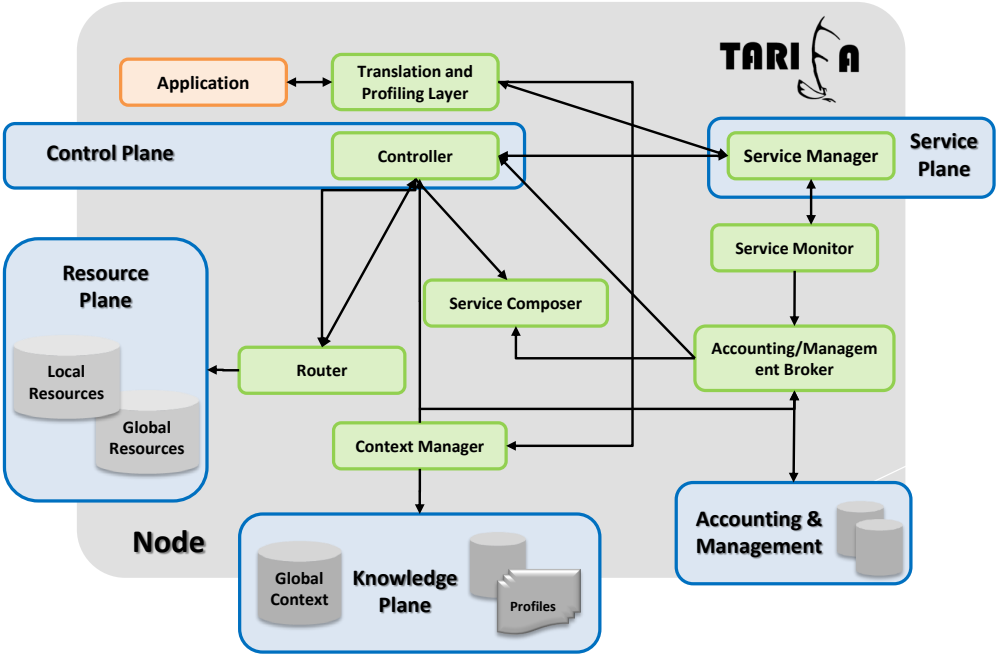


FIGURE A.2: TARIFA-compliant Node Architecture.

# Bibliography


[1] Framework DDP BA TMF518 FMW Version 1.2. Document Delivery Package, TeleManagement Forum, September 2011.

[2] D. Meyer. The Locator Identifier Separation Procotol (LISP). http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_11-1/111_lisp.html, 2011.

[3] D. C. Wimalasuriya and D. Dou. Ontology-based Information Extraction: An Introduction and a Survey of Current Approaches. *J. Inf. Sci.*, 36(3):306–323, June 2010.

[4] D. Meyer, L. Zhang, and K. Fall. Report from the IAB Workshop on Routing and Addressing. RFC 4984, IETF, September 2007. https://tools.ietf.org/html/rfc4984.

[5] S. Bishop, M. Fairbairn, M. Norrish, P. Sewell, M. Smith, and K. Wansbrough. Rigorous Specification and Conformance Testing Techniques for Network Protocols, As Applied to TCP, UDP, and Sockets. In *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '05, pages 265–276, New York, NY, USA, 2005. ACM.

[6] J.E. López de Vergara, V.A. Villagra, J.I. Asensio, and J. Berrocal. Ontologies: giving semantics to network management models. *Network, IEEE*, 17(3):15–21, May 2003.

[7] J. Schonwalder, M. Bjorklund, and P. Shafer. Network configuration management using NETCONF and YANG. *Communications Magazine, IEEE*, 48(9):166–173, Sept 2010.

[8] F. Le, S. Lee, T. Wong, H.S. Kim, and D. Newcomb. Detecting Network-Wide and Router-Specific Misconfigurations Through Data Mining. *Networking, IEEE/ACM Transactions on*, 17(1):66–79, Feb 2009.

[9] A. Pras, J. Schonwalder, M. Burgess, O. Festor, G.M. Perez, R. Stadler, and B. Stiller. Key Research Challenges in Network Management. *Communications Magazine, IEEE*, 45(10):104–110, 2007.

[10] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman. Network Configuration Protocol (NETCONF). RFC 6241, IETF, June 2011. http://tools.ietf.org/html/rfc6241.

[11] A Martinez, M. Yannuzzi, V. López, D. López, W. Ramírez, R. Serral-Gracia, X. Masip-Bruin, M. Maciejewski, and J. Altmann. Network Management Challenges and Trends in Multi-Layer and Multi-Vendor Settings for Carrier-Grade Networks. *Communications Surveys Tutorials, IEEE*, 16(4):2207–2230, Fourthquarter 2014.

[12] C. Chappell. The Business Case for NETCONF/YANG in Network Devices. White Paper, Heavy Reading, October 2013.

[13] M. Bjorklund. YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF). RFC 6020, IETF, October 2010. http://www.ietf.org/rfc/rfc6020.txt.

[14] IETF. NETCONF Data Modeling Language (NETMOD). http://datatracker.ietf.org/wg/netmod/documents/.

[15] E. Nordmark and M. Bagnulo. Shim6: Level 3 multihoming shim protocol for ipv6. RFC 5533, IETF, June 2009.

[16] C. Vogt. Six/one router: A scalable and backwards compatible solution for provider-independent addressing. In *Proceedings of the 3rd International Workshop on Mobility in the Evolving Internet Architecture*, MobiArch '08, pages 13–18, New York, NY, USA, 2008. ACM.

[17] X. Yang and X. Ji. Host identity protocol realizing the separation of the location and host identity. In *Information and Automation, 2008. ICIA 2008. International Conference on*, pages 749–752, June 2008.

[18] C. Paasch and O. Bonaventure. Multipath tcp. *Queue*, 12(2):40:40–40:51, February 2014.

[19] D. Massey, L. Wang, B. Zhang, and L. Zhang. A Proposal for Scalable Internet Routing and Addressing. Internet-Draft Informational, IETF, February 2007.

[20] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis. The Locator/ID Separation Protocol (LISP). RFC 6830, IETF, December 2013.

[21] B. Quoitin, L. Iannone, C. de Launois, and O. Bonaventure. Evaluating the Benefits of the Locator/Identifier Separation. In *Proceedings of MobiArch (ACM SIGCOMM Workshop)*, Kyoto, Japan, August 2007.

[22] V. Fuller, D. Farinacci, D. Meyer, and D. Lewis. LISP Alternative Topology (LISP+ALT). Internet-Draft, IETF, December 2011.

[23] E. Lear. NERD: A Not-so-novel Endpoint ID (EID) to Routing Locator (RLOC) Database. RFC 6837, IETF, January 2013.

[24] V. Fuller and D. Farinacci. Locator/ID Separation Protocol (LISP) Map-Server Interface. RFC 6833, IETF, January 2013.

[25] M. Yannuzzi, X. Masip-Bruin, E. Grampin, R. Gagliano, A. Castro, and M. German. Managing interdomain traffic in Latin America: a new perspective based on LISP. *Communications Magazine, IEEE*, 47(7):40–48, July 2009.

[26] GENI: Global Environment for Network Innovation. http://www.geni.net/, 2015.

[27] The FP7 4WARD Project. http://www.4ward-project.eu/, 2008-2010.

[28] T. Koponen, M. Chawla, B. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica. A data-oriented (and beyond) network architecture. *SIGCOMM Comput. Commun. Rev.*, 37(4):181–192, August 2007.

[29] TRILOGY: Arquitecting the Future Internet. http://trilogy-project.org/, 2008-2010.

[30] V. Dimitrov and V. Koptchev. PSIRP Project – Publish-subscribe Internet Routing Paradigm: New Ideas for Future Internet. In *Proceedings of the 11th International Conference on Computer Systems and Technologies and Workshop for PhD Students in Computing on International Conference on Computer Systems and Technologies*, CompSysTech '10, pages 167–171, New York, NY, USA, 2010. ACM.

[31] TARIFA: The Atomic Redesign of the Internet Future Architecture. http://www.i2cat.net/en/projecte/tarifa-1, 2010-2011.

[32] The Internet Engineering Task Force Official Web Site. http://www.ietf.org/.

[33] The International Telecommunication Union Official Web Site. http://www.itu.int/.

[34] The Optical Internetworking Forum Official Web Site. http://www.oiforum.com/.

[35] Telemanagement Forum Official Web Site. http://www.tmforum.org/.

[36] F. Muñoz, V. López, O.G. de Dios, and J.P. Fernández-Palacios. Multi-layer restoration in hierarchical IP/MPLS over WSON networks. In *Networks and Optical Communications (NOC), 2012 17th European Conference on*, pages 1–6, June 2012.

[37] Juniper PTX Official Site. http://www.juniper.net/us/en/dm/supercore/.

[38] Huawei's OptiX OSN 7500 II. http://www.huawei.com/en/products/transport-network/hybrid-mstp/osn7500II/index.htm.

[39] A. Farrel, J. P. Vasseur, and J. Ash. A Path Computation Element-Based Architecture. RFC 4655, IETF, August 2006. http://tools.ietf.org/rfc/rfc4655.txt.

[40] M. Yannuzzi, X. Masip Bruin, Ó. González de Dios, C. García Argos, M. Maciejewski, and J. Altmann. Bridging the Interoperability Gap Between the Internet and Optical Network Management Systems. In *Network and Optical Communications (NOC) Conference*, 2011.

[41] Infonetics Research - Service Provider Routers and Switches report. http://www.infonetics.com/pr/2013/2Q13-Service-Provider-Routers-Switches-Market-Highlights.asp.

[42] A. Gupta. Network Management: Current Trends and Future Perspectives. *Journal of Network and Systems Management*, 14(4):483–491, December 2006.

[43] T. McElligott. The challenge of managing multivendor networks. *Billing and OSS World*, 2008.

[44] J. Case, M. Fedor, M. Schoffstall, and J. Davin. Simple Network Management Protocol (SNMP). RFC 1157, Internet Engineering Task Force, May 1990.

[45] D. Mauro and K. Schmidt. *Essential SNMP, 2nd Edition*. O'Reilly Media, 2005.

[46] Telemanagement Forum MTOSI Web Page. Technical report.

[47] JUNOS OS NETCONF XML Management Protocol Guide, Release 11.4. Technical Documentation, JUNIPER, October 2011.

[48] Cisco Networking Services Configuration Guide - Network Configuration Protocol. Technical Documentation, CISCO, March 2013.

[49] Tail-f Systems: Build On-Device Management Applications with ConfD. http://www.tail-f.com/on-device-configuration-management/.

[50] S. Chisholm, A. Clemm, and J. Tjong. Using XML Schema to define NETCONF Content. Internet-Draft, Network Working Group, 2008.

[51] H. Cui, B. Zhang, G. Li, X. Gao, and Y. Li. Contrast Analysis of NETCONF Modeling Languages: XML Schema, Relax NG and YANG. In *Communication Software and Networks, 2009. ICCSN '09. International Conference on*, pages 322–326, 2009.

[52] L. Johansson. NETCONF Configuration Data Modeling using OWL. Internet-draft, Internet Engineering Task Force (IETF), 2008.

[53] H. Xu and D. Xiao. Data modeling for NETCONF-based network management: XML schema or YANG. In *Communication Technology, 2008. ICCT 2008. 11th IEEE International Conference on*, pages 561–564, 2008.

[54] OWL Web Ontology Language Overview. http://www.w3.org/TR/owl-features/.

[55] D. Caldwell, A. Gilbert, J. Gottlieb, A. Greenberg, G. Hjalmtysson, and J. Rexford. The Cutting EDGE of IP Router Configuration. *SIGCOMM Comput. Commun. Rev.*, 34(1):21–26, January 2004.

[56] Cisco Systems. Cisco active network abstraction 3.7.2 theory of operations guide. Technical report, Cisco Systems, 2012.

[57] E. Mannie. Generalized Multiprotocol Label Switching (GMPLS) Architecture. RFC 3945, IETF, October 2004.

[58] ITU-T. *G.8080/Y.1304: Architecture for the automatically switched optical network (ASON)*. International Telecommunications Union, 2012.

[59] IPoDWDM. http://www.cisco.com/en/US/netsol/ns1192/networking_solutions_solution.html.

[60] Juniper Networks. White paper: Improving network efficiency, reliability, and operations with IPoDWDM. Technical report, Juniper Networks, 2010.

[61] NTT Com Demonstrates GMPLS over 40 Gbps Network. http://www.ntt.com/release_e/news06/0006/r_0612.html, June 2006.

[62] S. Das, G. Parulkar, and N. McKeown. Why OpenFlow/SDN Can Succeed Where GMPLS Failed. In *European Conference and Exhibition on Optical Communication*, volume 1, 2012.

[63] International Telecommunication Union (ITU). Recommendation G.698.2: Amplified multichannel dense wavelength division multiplexing applications with single channel optical interfaces, November 2009.

[64] CyMS Multi-Layer Management System. http://cyaninc.com/cyms/cyan-cyms.

[65] M. Bocci, S. Bryant, D. Frost, L. Levrau, and L. Berger. A Framework for MPLS in Transport Networks. RFC 5921, Internet Engineering Task Force (IETF), July 2010. http://tools.ietf.org/html/rfc5921.

[66] K. Shiomoto and et al. Requirements for GMPLS-based multi-region and multi-layer networks (MRN/MLN). RFC 5212, IETF, July 2008.

[67] MPLS Working Group. http://datatracker.ietf.org/wg/mpls/.

[68] L. Lei and S. Sampalli. Distributed Online LSP Merging Algorithms for MPLS-TE. In Hermann Meer and Nina Bhatti, editors, *Quality of Service IWQoS 2005*, volume 3552 of *Lecture Notes in Computer Science*, pages 366–368. Springer Berlin Heidelberg, 2005.

[69] L. Lobo. *MPLS Configuration on Cisco IOS Software*. Cisco Press, 2005.

[70] G. Swallow, S. Bryant, and L. Andersson. Avoiding Equal Cost Multipath Treatment in MPLS Networks. RFC 4928, Network Working Group, June 2007.

[71] I. Busi and D. Allan. Operations, Administration, and Maintenance Framework for MPLS-Based Transport Networks. RFC 6371, Internet Engineering Task Force (IETF), 2011. http://tools.ietf.org/html/rfc6371.

[72] B. Niven-Jenkins, D. Brungard, M. Betts, N. Sprecher, and S. Ueno. Requirements of an MPLS Transport Profile. RFC 5654, Network Working Group, 2009. http://tools.ietf.org/html/rfc5654.

[73] M. Vigoureux, D. Ward, and M. Betts. Requirements for Operations, Administration, and Maintenance (OAM) in MPLS Transport Networks. RFC 5860, Internet Engineering Task Force (IETF), May 2010. http://tools.ietf.org/html/rfc5860.

[74] L. Andersson, L. Berger, L. Fang, N. Bitar, and E. Gray. MPLS Transport Profile (MPLS-TP) Control Plane Framework. RFC 6373, Internet Engineering Task Force (IETF), 2011. http://tools.ietf.org/html/rfc6373.

[75] S. Hubbard. MPLS-TP in Next-Generation Transport Networks. White paper, Light Reading/Heavy Reading MPLS-TP Initiative, July 2011.

[76] Ethernet vs. MPLS-TP in the Access, 2012. http://www.rad.com/12/Ethernet-MPLS-TP-in-the-Access/23979/.

[77] J. Yang and S.J. B. Yoo. An integrated network management system with online traffic engineering for optical label switching networks. In *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, volume 3, nov.-3 dec. 2004.

[78] L. Raptis, G. Hatzilias, F. Karayannis, K. Vaxevanakis, and E. Grampin. An integrated network management approach for managing hybrid IP and WDM networks. *Network, IEEE*, 17(3):37 – 43, May-June 2003.

[79] WDM and IP Network MANagement (WINMAN). http://winman.upc.es/, 2001-2003.

[80] N. Vardalachos, E. Grampin, A. Galis, and J. Serrat. Policy Management Approach for IP over WDM Networks: A Synthesis Study. In *Annual London Conference on Communications*, 2001.

[81] Recommendation ITU-T G.800: Unified functional architecture of transport networks. ITU-T G.800, ITU-T, February 2012. http://www.itu.int/rec/T-REC-G.800-201202-I/es.

[82] R. Cafini, W. Cerroni, C. Raffaelli, and M. Savi. Standard-based approach to programmable hybrid networks. *Communications Magazine, IEEE*, 49(5):148–155, May 2011.

[83] H. Takenouchi, R. Urata, T. Nakahara, T. Segawa, H. Ishikawa, and R. Takahashi. First demonstration of a prototype hybrid optoelectronic router. In *Optical Communication, 2009. ECOC '09. 35th European Conference on*, volume 2009-Supplement, pages 1–2, 2009.

[84] E. Oki, T. Takeda, JL. Le Roux, and A. Farrel. Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering. RFC 5623, IETF, September 2009. http://tools.ietf.org/rfc/rfc5623.txt.

[85] JP. Vasseur and J.L. Le Roux A. Path Computation Element Communication Protocol. RFC 5440, IETF, March 2009. http://tools.ietf.org/rfc/rfc5440.txt.

[86] IETF PCE Working Group Web Page. http://datatracker.ietf.org/wg/pce/charter/.

[87] V. López, B. Huiszoon, J. Fernández-Palacios, O. González de Dios, and J. Aracil. Path Computation Element in Telecom Networks: Recent developments and standardization activities. In *Optical Network Design and Modeling (ONDM), 2010 14th Conference on*, pages 1–6, 2010.

[88] C. Margaria, O. González de Dios, and F. Zhang. PCEP extensions for GMPLS. IETF Internet Draft, March 2012. http://datatracker.ietf.org/doc/draft-ietf-pce-gmpls-pcep-extensions/.

[89] Y. Lee, G. Bernstein, J. Martensson, T. Takeda, T. Tsuritani, and O. González de Dios. PCEP Requirements for WSON Routing and Wavelength Assignment. IETF Internet Draft, April 2012. http://datatracker.ietf.org/doc/draft-ietf-pce-wson-routing-wavelength/.

[90] JP. Vasseur and et al. A Backward-Recursive PCE-Based Computation (BRPC) Procedure to Compute Shortest Constrained Inter-Domain Traffic Engineering Label Switched Paths. RFC 5441, IETF, April 2009. http://tools.ietf.org/rfc/rfc5441.txt.

[91] F. Cugini, A. Giorgetti, N. Andriolli, I. Paolucci, L. Valcarenghi, and P. Castoldi. Multiple Path Computation Element (PCE) Cooperation for Multi-layer Traffic Engineering. In *Optical Fiber Communication and the National Fiber Optic Engineers Conference, 2007. OFC/NFOEC 2007. Conference on*, pages 1–3, 2007.

[92] M. Chamania, O. González de Dios, V. López, M. Drogon M. Cuaresma, A. Jukan, X. Masip-Bruin, and M. Yannuzzi. Coordinated Computation of Multi-layer Paths via Inter-layer PCE Communication: Standards, Interoperability and Deployment. In *IEEE ICC 2012 WS - From Research to Standards*, 2012.

[93] A. Bukva, R. Casellas, R. Martínez, and R. Muñoz. A Dynamic Path-Computation Algorithm for a GMPLS-Enabled Multi-layer Network. 4:436–448, June 2012.

[94] B. Jabbari, S. Gong, and E. Oki. On Constraints for Path Computation in Multilayer Switched Networks. In *IEICE Transactions on Communications*, volume E90-B, pages 1922–1927, August 2007.

[95] S. Gong and B. Jabbari. Optimal and Efficient End-to-End Path Computation in Multi-Layer Networks. In *Communications, 2008. ICC '08. IEEE International Conference on*, pages 5767–5771, 2008.

[96] X. Yang, T. Lehman, K. Ogaki, and T. Otani. A study on cross-layer multi-constraint path computation for IP-over-optical networks. In *Proceedings of the 2009 IEEE international conference on Communications*, ICC'09, pages 2247–2252, Piscataway, NJ, USA, 2009. IEEE Press.

[97] P. Fodor, G. Enyedi, G. Rétvári, and T. Cinkler. Layer-preference policies in multi-layer GMPLS networks. In *Photonic Network Commun.*, volume 18, pages 300–313, February 2009.

[98] F. Dijkstra, J.V.D. Ham, P. Grosso, and C. de Laat. A path finding implementation for multi-layer networks. *Future Generation Computer Systems*, 25(2):142 – 146, 2009.

[99] F. Kuipers and F. Dijkstra. Path selection in multi-layer networks. *Comput. Commun.*, 32(1):78–85, January 2009.

[100] A. Rahat Rubuyat. Path Computation Element in GMPLS Enabled Multi-layer Networks. Masters' Degree Project, KTH Electrical Engineering, 2006.

[101] D. Eppstein. Finding the k shortest paths. *SIAM J. Comput.*, 28(2):652–673, February 1999.

[102] V. M. J. Pelayo and A. M. Varó. Computing the k shortest paths: A new algorithm and an experimental comparison. In *Proceedings 3rd International Workshop Algorithm Engineering*, number 1668, pages 15–29, 1999.

[103] E. Q. V. Martins and M. M. B. Pascoal. A new implementation of YEN's ranking loopless paths algorithm. 1:121–134, January 2003.

[104] H. Zhu, H. Zhang, K. Zhu, and B. Mukherjee. A novel generic graph model for traffic grooming in heterogeneous WDM networks. In *IEEE/ACM Transactions on Networking*, volume 11, pages 258–299, 2003.

[105] W. Yao and B. Ramamurthy. A link budled auxiliary graph model for constrained dynamic traffic grooming in WDM mesh networks. In *IEEE Journal on Selected Areas in Communications*, volume E90-B, pages 1922–1927, August 2007.

[106] JL. Le Roux, JP. Vasseur, and Y. Lee. Encoding of Objective Functions in the Path Computation Element Communication Protocol (PCEP). RFC 5541, Network Working Group, June 2009.

[107] F. Zhang and A. Farrel. Conveying Vendor-Specific Constraints in the Path Computation Element Protocol. Internet draft, Network Working Group, April 2013.

[108] D. Papadimitriou and et al. Generalized Multi-Protocol Label Switching (GMPLS) Protocol Extensions for Multi-layer and Multi-Region Networks (MLN/MRN). RFC 6001, IETF, October 2010.

[109] J. Lang. Link Management Protocol (LMP). RFC 4204, Network Working Group, October 2005.

[110] A. Farrel and I. Bryskin. *GMPLS - Architecture and Applications*. Morgan Kaufmann, 2006.

[111] EU Project ONE. http://www.ict-one.eu, 2010-2013.

[112] Open Networking Foundation. OpenFlow Switch Specification. Version 1.1.0. http://www.openflow.org/documents/openflow-spec-v1.1.0.pdf.

[113] J. Schoenwaelder. Overview of the 2002 IAB Network Management Workshop. RFC 3535, IETF, May 2003. http://tools.ietf.org/search/rfc3535#page-10.

[114] NETCONF Data Modeling Language (netmod) active internet drafts. http://datatracker.ietf.org/wg/netmod/, 2014.

[115] P.C.G. Costa, C. d'Amato, N. Fanizzi, K.B. Laskey, K.J. Laskey, M. Nickles, and M. Pool. Towards Machine Learning on the Semantic Web. In *Uncertainty Reasoning for the Semantic Web I - ISWC International Workshop*, volume 5327 of *Lecture Notes in Computer Science*, pages 282–314, Piscataway, NJ, USA, 2008. URSW.

[116] A. Ka Yiu Wong, P. Ray, N. Parameswaran, and J. Strassner. Ontology Mapping for the Interoperability Problem in Network Management. *IEEE Journal on Selected Areas in Communications*, 23(10):2058–2068, October 2005.

[117] H. Xu and D. Xiao. Applying Semantic Web Services to Automate Network Management. In *2nd IEEE Conference on Industrial Electronics and Applications, 2007. ICIEA*, pages 461–466, May 2007.

[118] H. Xu and D. Xiao. A Common Ontology-Based Intelligent Configuration Management Model for IP Network Devices. In *Innovative Computing, Information and Control, 2006. ICICIC '06. First International Conference on*, volume 1, pages 385–388, Aug 2006.

[119] J. López de Vergara, V. Villagri, J. Asensio, and J. Berrocal. Ontologies: Giving Semantics to Network Management Models. *IEEE Network*, 17(3):15–21, May 2003.

[120] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. http://www.w3.org/Submission/SWRL/.

[121] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, and et al. OWL-S: Semantic markup for web services, 2004. http://www.w3.org/Submission/OWL-S/.

[122] P. Pan and T. Nadeau. Software-Defined Network (SDN) Problem Statement and Use Cases for Data Center Applications. draft-pan-sdn-dc-problem-statement-and-use-cases, IETF, March 2012.

[123] S. Das, G. Parulkar, and N. McKeown. Unifying Packet and Circuit Switched Networks. In *Below IP Networking workshop in conjunction with Globecom '09*, 2009.

[124] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar. Architectural Guidelines for Multipath TCP Development. RFC 6182, Internet Engineering Task Force, March 2011.

[125] O. Bonaventure, M. Handley, and C. Raiciu. An overview of Multipath TCP. *USENIX login;*, October 2012.

[126] R. Van der Pol, M. Bredely, A. Barczyky, B. Overeinderz, N. van Adrichemx, and F. Kuipersx. Experiences with MPTCP in an intercontinental OpenFlow network. In *The 29th TERENA Networking Conference*, pages 1617–1624, June 2013.

[127] R. Van der Pol, S. Boele, F. Dijkstra, A. Barczyk, G. van Malenstein, J.H. Chen, and J. Mambretti. Multipathing with MPTCP and OpenFlow. In *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:*, pages 1617–1624, November 2012.

[128] F. Németh, B. Sonkoly, L. Csikor, and A. Gulyás. A Large-scale Multipath Playground for Experimenters and Early Adopters. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 481–482, New York, USA, 2013. ACM.

[129] M. Coudron, S. Secci, G. Pujolle, P. Raad, and P. Gallard. Cross-layer cooperation to boost multipath TCP performance in cloud networks. In *Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on*, pages 58–66, November 2013.

[130] P. Gorja and R. Kurapati. Extending open vSwitch to L4-L7 service aware OpenFlow switch. In *Advance Computing Conference (IACC), 2014 IEEE International*, pages 343–347, February 2014.

[131] S. Gringeri, N. Bitar, and T.J. Xia. Extending software defined network principles to include optical transport. *Communications Magazine, IEEE*, 51(3):32–40, March 2013.

[132] B. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti. A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks. *Communications Surveys Tutorials, IEEE*, PP(99):1–18, 2014.

[133] A. Sadasivarao, S. Syed, P. Pan, C. Liou, A. Lake, C. Guok, and I. Monga. Open Transport Switch: A Software Defined Networking Architecture for Transport Networks. In *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, HotSDN '13, pages 115–120, New York, USA, 2013. ACM.

[134] K. Idoudi and H. Elbiaze. Enhancing OpenFlow to enable multilayer networking. In *Transparent Optical Networks (ICTON), 2013 15th International Conference on*, pages 1–5, June 2013.

[135] M. Channegowda, R. Nejabati, and D. Simeonidou. Software-defined optical networks technology and infrastructure: Enabling software-defined optical network operations. *Optical Communications and Networking, IEEE/OSA Journal of*, 5(10):A274–A282, October 2013.

[136] Network Functions Virtualisation Introductory White Paper. White paper, October 2012.

[137] PTX Series Packet Transport Switch Datasheet. http://www.juniper.net/us/en/local/pdf/datasheets/1000364-en.pdf, December 2011.

[138] M. Yannuzzi, X. Masip-Bruin, and O. Bonaventure. Open issues in interdomain routing: a survey. *Network, IEEE*, 19(6):49–56, Nov 2005.

[139] X. Zhao, D.J. Pacella, and J. Schiller. Routing Scalability: An Operator's View. *Selected Areas in Communications, IEEE Journal on*, 28(8):1262–1270, October 2010.

[140] L. Cittadini, W. Muhlbauer, S. Uhlig, R. Bush, P. François, and O. Maennel. Evolution of Internet Address Space Deaggregation: Myths and Reality. *Selected Areas in Communications, IEEE Journal on*, 28(8):1238–1249, October 2010.

[141] Endpoints and endpoint names: A Proposed Enhancement to the Internet Architecture. http://www.chiappa.net/~jnc/tech/endpoints.txt, 1999.

[142] J. Saltzer. On the naming and binding of network destinations, 1993.

[143] R. Ahmed, R. Boutaba, F. Cuervo, Y. Iraqi, Tianshu Li, N. Limam, Jin Xiao, and J. Ziembicki. Service naming in large-scale and multi-domain networks. *Communications Surveys Tutorials, IEEE*, 7(3):38–54, Third 2005.

[144] P. Leach, M. Mealling, and R. Salz. A Universally Unique IDentifier (UUID) URN Namespace. RFC 4122, IETF, July 2005. https://tools.ietf.org/html/rfc4122.

[145] T. Berners-Lee, L. Masinter, and M. McCahill. Uniform Resource Locators (URL). RFC 1738, IETF, December 1994. https://www.ietf.org/rfc/rfc1738.txt.

[146] K. Bontcheva, H. Cunningham, A. Kiryakov, and V. Tablan. Semantic Annotation and Human Language Technology. In *Semantic Web Technologies: Trends and Research in Ontology-based Systems*, pages 29–50. Wiley, 2006.

[147] S. Sarawagi. Information Extraction. *Found. Trends databases*, 1(3):261–377, March 2008.

[148] R. Studer, V.R. Benjamins, and D. Fensel. Knowledge Engineering: Principles and Methods. *Data and Knowledge Engineering*, 25(1–2):161 – 197, 1998.

[149] X. Han and J. Zhao. Structural Semantic Relatedness: A Knowledge-based Method to Named Entity Disambiguation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 50–59, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[150] Einat Minkov and William W. Cohen. Graph based similarity measures for synonym extraction from parsed text. In *Workshop Proceedings of TextGraphs-7 on Graph-based Methods for Natural Language Processing*, TextGraphs-7 '12, pages 20–24, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[151] Lushan Han, Tim Finin, Paul McNamee, Anupam Joshi, and Yelena Yesha. Improving word similarity by augmenting pmi with estimates of word polysemy. *IEEE Transactions on Knowledge and Data Engineering*, 25(6):1307–1322, 2013.

[152] M. Han. Semantic information retrieval based on wikipedia taxonomy. *International Journal of Computer Applications Technology and Research*, 2(1):77 – 80, 2013.

[153] A. Hliaoutakis, G. Varelas, E. Voutsakis, E. G. M. Petrakis, and E. Milios. Information Retrieval by Semantic Similarity. In *Intern. Journal on Semantic Web and Information Systems (IJSWIS), 3(3):55–73, July/Sept. 2006. Special Issue of Multimedia Semantics*, 2006.

[154] G. Varelas, E. Voutsakis, E. G. M. Petrakis, E. E. Milios, and P. Raftopoulou. Semantic Similarity Methods in WordNet and their Application to Information Retrieval on the Web. In *In: 7 th ACM Intern. Workshop on Web Information and Data Management (WIDM 2005*, pages 10–16. ACM Press, 2005.

[155] I. Hendrickx, W. Daelemans, E. Marsi, and E. Krahmer. Reducing Redundancy in Multi-document Summarization Using Lexical Semantic Similarity. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation*, UCNLG+Sum '09, pages 63–66, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[156] Z. Zhang, A. Gentile, and F. Ciravegna. Recent advances in methods of lexical semantic relatedness – a survey. *Natural Language Engineering*, 19:411–479, 10 2013.

[157] L. Zhang, J. Hu, and X. Zheng. Measuring semantic relatedness based on ontology. In *Automatic Control and Artificial Intelligence (ACAI 2012), International Conference on*, pages 1335–1338, March 2012.

[158] P. Buitelaar, P. Cimiano, A. Frank, M. Hartung, and S. Racioppa. Ontology-based information extraction and integration from heterogeneous data sources. *Int. J. Hum.-Comput. Stud.*, 66(11):759–788, November 2008.

[159] Y. Wei-Guo, Y. Ling-Wei, L. Ya-Qing, and L. Zhi. An Ontology-Based Web Information Extraction Approach. In *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, volume 1, pages V1–132–V1–136, May 2010.

[160] D. Maynard, M. Yankova, R. Kourakis, and A. Kokossis. Ontology-based information extraction for market monitoring and technology watch. In *In ESWC Workshop "End User Apects of the Semantic Web*, 2005.

[161] K. Nebhi. Ontology-based information extraction for french newspaper articles. In Birte Glimm and Antonio Krüger, editors, *KI 2012: Advances in Artificial Intelligence*, volume 7526 of *Lecture Notes in Computer Science*, pages 237–240. Springer Berlin Heidelberg, 2012.

[162] K. Nebhi. *Ontology-Based Information Extraction from Twitter*, pages 17–22. Proceedings of the Workshop on Information Extraction and Entity Analytics on Social Media Data - COLING 2012. The COLING 2012 Organizing Committee, 2012. ID: unige:24683.

[163] D. Çelik and A. Elçi. An ontology-based information extraction approach for résumés. In Q. Zu, B. Hu, and A. Elçi, editors, *Pervasive Computing and the Networked World*, volume 7719 of *Lecture Notes in Computer Science*, pages 165–179. Springer Berlin Heidelberg, 2013.

[164] E. Soysal, I. Cicekli, and N. Baykal. Design and evaluation of an ontology based information extraction system for radiological reports. *Computers in Biology and Medicine*, 40(11–12):900 – 911, 2010.

[165] ONDC: Ontology for Network Device Configuration. http://www.netit.upc.edu/ondc, 2014.

[166] Open Networking Foundation. OpenFlow Management and Configuration Protocol 1.2. www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow-config/of-config-1.2.pdf.

[167] H. Kim and N. Feamster. Improving network management with software defined networking. *Communications Magazine, IEEE*, 51(2):114–119, February 2013.

[168] D. Le Goff. SDN Challenges Discussed at Carrier Cloud Summit. http://www.6wind.com/blog/sdn-challenges-discussed-at-carrier-cloud-summit/, 6WIND.

[169] S. Vissicchio, L. Vanbever, and O. Bonaventure. Opportunities and Research Challenges of Hybrid Software Defined Networks. *ACM Computer Communication Review*, 44(2), April 2014.

[170] D. Levin, M. Canini, S. Schmid, and A. Feldmann. Incremental SDN Deployment in Enterprise Networks. In *SIGCOMM'13 Demo*, Hong Kong, China, August 2013.

[171] M. Grassi, M. Nucci, and F. Piazza. Towards a semantically-enabled holistic vision for energy optimisation in smart home environments. In *IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pages 299–304, April 2011.

[172] M. Nucci, M. Grassi, and F. Piazza. Ontology-Based Device Configuration and Management for Smart Homes. In *Neural Nets and Surroundings*, volume 19 of *Smart Innovation, Systems and Technologies*, pages 301–310. Springer Berlin Heidelberg, 2013.

[173] H. Wicaksono, V. Schubert, S. Rogalski, Y.Ait Laydi, and J. Ovtcharova. Ontology-driven Requirements Elicitation in Product Configuration Systems. In Hoda A. ElMaraghy, editor, *Enabling Manufacturing Competitiveness and Economic Sustainability*, pages 63–67. Springer Berlin Heidelberg, 2012.

[174] F. Colace, M. De Santo, and P. Napoletano. Product Configurator: An Ontological Approach. In *Ninth International Conference on Intelligent Systems Design and Applications, 2009. ISDA '09*, pages 908–912, Nov 2009.

[175] M. Dong, D. Yang, and L. Su. Ontology-based service product configuration system modeling and development. *Expert Systems with Applications*, 38(9):11770 – 11786, 2011.

[176] D. Yang, R. Miao, H. Wu, and Y. Zhou. Product configuration knowledge modeling using ontology web language. *Expert Systems with Applications*, 36(3, Part 1):4399 – 4411, 2009.

[177] D. Yang, M. Dong, and R. Miao. Development of a product configuration system with an ontology-based approach. *Computer-Aided Design*, 40(8):863 – 878, 2008.

[178] J. E. López de Vergara, V. Villagrá, and J. Berrocal. Application of OWL-S to Define Management Interfaces Based on Web Services. In Jordi Dalmau Royo and Go Hasegawa, editors, *Management of Multimedia Networks and Services*, volume 3754 of *LNCS*, pages 242–253. Springer, 2005.

[179] A. Wong, P. Ray, N. Parameswaran, and J. Strassner. Ontology mapping for the interoperability problem in network management. *Selected Areas in Communications, IEEE Journal on*, 23(10):2058–2068, 2005.

[180] N. Lasierra, A. Alesanco, D. O'Sullivan, and J. García. An autonomic ontology-based approach to manage information in home-based scenarios: From theory to practice. *Data and Knowledge Engineering*, 87(0):185 – 205, 2013.

[181] J. Keeney, S. van der Meer, and G. Hogan. A recommender-system for telecommunications network management actions. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, pages 760–763, May 2013.

[182] J. López, V. Villagrá, and J. Berrocal. Applying the web ontology language to management information definitions. *Communications Magazine, IEEE*, 42(7):68–74, 2004.

[183] J. López de Vergara, A. Guerrero, V. Villagrá, and J. Berrocal. Ontology-Based Network Management: Study Cases and Lessons Learned. *Journal of Network and Systems Management*, 17(3):234–254, 2009.

[184] J. E. López de Vergara, V. A. Villagrá, C. Fadón, J. M. González, J. A. Lozano, and M. Álvarez Campana. An autonomic approach to offer services in OSGi-based home gateways. *Computer Communications*, 31(13):3049 – 3058, 2008. Special Issue: Self-organization and self-management in communications as applied to autonomic networks.

[185] OWL Web Ontology Language API. http://owlapi.sourceforge.net/.

[186] Z. Li and K. Ramani. Ontology-based design information extraction and retrieval. *AI EDAM: Artificial Intelligence for Engineering Design, Analysis, and Manufacturing*, 21:137–154, 4 2007.

[187] Stanford CoreNLP. http://nlp.stanford.edu/software/corenlp.shtml.

[188] Apache OpenNLP. https://opennlp.apache.org/ .

[189] A. Esuli and F. Sebastiani. Evaluating information extraction. In Maristella Agosti, Nicola Ferro, Carol Peters, Maarten de Rijke, and Alan Smeaton, editors, *Multilingual and Multimodal Information Access Evaluation*, volume 6360 of *Lecture Notes in Computer Science*, pages 100–111. Springer Berlin Heidelberg, 2010.

[190] D. Maynard. Metrics for evaluation of ontology-based information. In *In WWW 2006 Workshop on "Evaluation of Ontologies for the Web*, 2006.

[191] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. Greenwood, H. Saggion, J. Petrak, Y. Li, and W. Peters. *Text Processing with GATE (V.6)*. 2011.

[192] D. Maynard, W. Peters, and Y. Li. Evaluating evaluation metrics for ontology-based applications: Infinite reflection. In *LREC*, 2008.

[193] G. Tsatsaronis, I. Varlamis, and M. Vazirgiannis. Text Relatedness Based on a Word Thesaurus. *J. Artif. Int. Res.*, 37(1):1–40, January 2010.

[194] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4. RFC 4271, IETF, January 2006. http://www.ietf.org/rfc/rfc4271.txt.

[195] G. Huston. The growth of the BGP table - 1994 to present. http://bgp.potaroo.net, 2014.

[196] Tian Bu, Lixin Gao, and Don Towsley. On characterizing bgp routing table growth. *Comput. Netw.*, 45(1):45–54, May 2004.

[197] R. Oliveira, R. Izhak-Ratzin, B. Zhang, and L. Zhang. Measurement of highly active prefixes in BGP. In *Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, volume 2, pages 5 pp.–, Nov 2005.

[198] A. Flavel, M. Roughan, N. Bean, and O. Maennel. Modeling BGP Table Fluctuations. In *Proceedings of the 20th International Teletraffic Conference on Managing Traffic Performance in Converged Networks*, ITC20'07, pages 141–153, Berlin, Heidelberg, 2007. Springer-Verlag.

[199] The Hot Standby Router Protocol (HSRP). http://www.cisco.com/en/US/tech/tk648/tk362/tk321/tsd_technology_support_sub-protocol_home.html, 2014.

[200] H. Luo, Y. Qin, and H. Zhang. A dht-based identifier-to-locator mapping approach for a scalable internet. *Parallel and Distributed Systems, IEEE Transactions on*, 20(12):1790–1802, Dec 2009.

[201] World Population to 2300 - Official United Nations Estimates and Projections of World, Regional and National Population Size . http://www.i2cat.net/en/projecte/tarifa-1, 2010-2011.

[202] G. Cordasco, L. Gargano, M. Hammart, and V. Scarano. Degree-optimal deterministic routing for P2P systems. In *Computers and Communications, 2005. ISCC 2005. Proceedings. 10th IEEE Symposium on*, pages 158–163, June 2005.