# For a joint operator-centric approach to assessing network management effort

Marc Bruyere
IIJ Research Lab
(Japan)

Daphne Tuncer
Ecole des Ponts ParisTech
(France)

## ABSTRACT

Even for experts, networks are complicated to troubleshoot, manage, and plan. Adopting human-centered design methods has the potential to facilitate the creation of more manageable networks. In this paper, we investigate a quantitative method that involves defining and measuring the complexity of operating different types of architectures from the perspective of the space of network parameters that need to be monitored and configured. We present OPLEX, a novel framework based on the analysis of YANG data models of network implementations that enables operators to compare architecture options based on the dimension of the parameter space. The benefits of the proposed framework are illustrated in the use case of Internet Exchange Point (IXP) network architectures, for which we take advantage of the rich set of publicly available data. We also exploit the survey results and direct consultations we conducted with operators and vendors of IXPs on their perception of complexity when operating different architectures. OPLEX is flexible, builds upon data models with widespread usage in the community, and provides a practical solution geared towards operators for characterizing the complexity of network architecture options.

## CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**; • **General and reference → Metrics**.

## KEYWORDS

Network management, automation, network operators, complexity

**ACM Reference Format:**
Marc Bruyere and Daphne Tuncer. 2023. For a joint operator-centric approach to assessing network management effort. In *Second Workshop on Situating Network Infrastructure with People, Practices, and Beyond (SNIP2+ '23), September 10, 2023, New York, NY, USA*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3609396.3610545

## 1 INTRODUCTION

Operators usually have the choice between different architecture options to implement their network. These options typically respond to two main needs: 1) deploying what is essential in terms of functionality to support the offered services, and 2) ensuring that the network can evolve in the face of future changes and / or unexpected events. From the perspective of an operator, the choice of an architecture has an impact on the efforts required to operate and manage the network. The type of architecture not only affects the time it takes to make changes and fix issues, it also drives the expertise and knowledge needed to perform network management tasks. Having in place mechanisms to evaluate the complexity of operating a network based on its architecture can thus support the operator in better understanding future expected efforts.

Different approaches were proposed to quantify the complexity associated with the operation and management of a network [18] [3][4]. A key contribution of previous work is the definition of quantifiable metrics enabling systematic comparison across network implementations [1][5]. In general, the factors of complexity in operating a network are based on three dimensions [2]: *i)* the operator, represented by their expertise; *ii)* the management interface, based on its degree of abstraction and automation; and *iii)* the network, described by the space of its configuration parameters.

In this paper, we revisit the link between evaluating the complexity of operating a network and characterizing the network through its parameter space, in the light of developments in the community towards the implementation of standardized models of network functionality. We present OPLEX (Operation comPLEXity), a framework based on the analysis of YANG [9] models of network architectures to determine and compare architecture options based on the dimension of the space of parameters that an operator needs to monitor and configure in order to manage underlying resources and services. In contrast to previous solutions, *e.g.,* [3][4], OPLEX is agnostic to an operator's internal standard specifics. It builds instead upon a standardized format with a widespread usage across vendors and operators [10]. In addition, it takes into account the whole functionality space of a network as enabled through the analysis of YANG models available for the full network stack at the device level, *i.e.,* from optical transport to routing policies and management (see OpenConfig [11] for example). Using OPLEX we develop a tool that can automatically extract the dimension of the network parameter space associated with a network architecture. Our tool is generic, *i.e.,* it can accommodate the YANG data models of any vendor implementation. It is also easily extensible, *i.e.,* new data models can easily be added to enrich the set of implementations for which complexity scores can be computed.

To illustrate the functionality of OPLEX, we elaborate on the complexity of operating a network in the specific use case of Internet eXchange Point (IXP) network architectures, for which we collected qualitative and quantitative datasets. In addition to the importance IXPs have today's in the Internet ecosystem [12][13], another motivation for focusing on this type of networks comes from the fact that IXPs have traditionally been engaged in an open

---

*SNIP2+ '23, September 10, 2023, New York, NY, USA*

ACM ISBN 979-8-4007-0304-1/23/09.
https://doi.org/10.1145/3609396.3610545

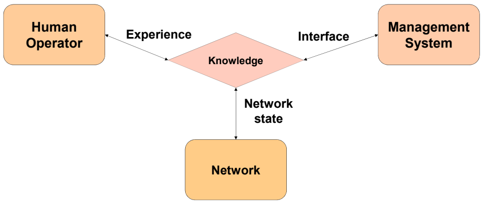

**Figure 1: Knowledge as a connector between the dimensions of the complexity of operating a network.**

approach to their business, as well as to their technical and performance specifications. Information about IXPs are publicly available from different sources on the web (*e.g.,* [14][15][16]), and individual webpages. Based on OPLEX tool, we compare the parameter space of network architectures used in today's IXPs. We discuss the implication on network complexity by putting the obtained results in perspective with a qualitative dataset of key operational considerations for the operators of these networks that we collected by disseminating a survey within the IXP community and engaging in direct consultations with multiple actors in this domain. Our results, based on responses received from eighteen IXPs and three leading vendors of the IXP market, reveal how IXP operators perceive the complexity and knowledge required to design and maintain nine selected architectures representing the up-to-date solutions to interconnect Internet autonomous networks.

This paper is a short version of a report made publicly available on ArXiv [7] in January 2022. The objective here is to share and discuss with the participants of the SNIP2+ workshop the ideas we have been developing over the last four years towards understanding the role of network operators in the automation of network management tasks. The approach we propose is work in-progress and the results are to be apprehended as initial reflective insights on studying network management complexity. We aim to expand our study by incorporating insights from sociology, as we recognize the significance of understanding the intricate interplay between people, practices, and other broader sociological factors in evaluating the complexity of network operation.

## 2 COMPLEXITY AND NETWORK PARAMETER SPACE

Various factors contribute to the complexity of operating a network. In a seminal paper [2] Behringer proposes to model factors of complexity based on three dimensions: the network, the management system and the (human) operator. As depicted in Fig. 1, *knowledge* acts as a connector between the three dimensions. More specifically, to operate the network, the operator builds upon knowledge of the environment, which gets enriched through experience. This knowledge can be described in terms of network state, *i.e.,* a set of parameters representing both for the hardware and software resources that need to be configured and/or monitored. Read and write access to that state is mediated through the management system that provides an interface between the network and the operator. The complexity of operating a network is a translation of that knowledge. It can be apprehended from three perspectives:

(1) by measuring the operator experience and level of expertise.
(2) by evaluating the degree of sophistication of the management interface, *i.e.,* support for abstraction, automation and intelligence.
(3) by quantifying the dimension of the network parameter space.

While the three dimensions are essential to the development of a comprehensive complexity assessment framework for network management, we focus on the first and the third dimension as a first step in this paper.

### 2.1 Network Functional Domains

Evaluating the complexity an operator faces operating and managing different types of network architectures through the dimension of the network parameter space is the underlying principle of most previously proposed approaches [3][4][17][18]. It comes from the observation that the larger the parameter space, the harder it is to maintain a full knowledge of that space, and hence to track, identify and correct operational issues [18].

In a networking context, the parameter space covers multiple functional domains, ranging from the physical infrastructure to the provided services, that can be classified in four main groups.

- Facility: all physical/virtual resources deployed for hosting and powering the network infrastructure.
- Interconnection: all physical/virtual resources and processes to enable local and remote connectivity across the network.
- Communication: all physical/virtual resources and processes to enable information exchange across the network.
- Services: all physical/virtual resources and processes to provide added value services on top of the infrastructure (*e.g.,* security)

Each group involves a variety of functional elements, including networking functions and equipment, for which different design and implementation choices can be selected, *e.g.,* layer-2 switching mechanisms, network operating system, switching platforms, *etc.* The combination of these functional elements define the parameter space of a network, and by extension, drives how complex it is to operate that network.

A key challenge to evaluate the complexity is to determine the granularity at which to take functional elements into account so as to be representative of the network operations. For instance, a naive approach would consist in assessing complexity as a count of the number of activated networking functions (*e.g.,* EVPN, MPLS, port filtering, *etc.*). Such an approach is oblivious to device and function state whereas this state is critical to network operations. In line with previous initiatives [3], we define in this work the functional elements at the granularity of network configurations. This approach offers a good trade-off between practicality, *i.e.,* it is easy for the operator to extract, and expressiveness, *i.e.,* it includes all parameters taken into account to reason upon network operations. As opposed to previous work, however, we investigate the use of standardized network data models.

### 2.2 Normalized Network Parameter Space

A main objective when evaluating the complexity of operating a network is to enable comparison between different network architectures. To provide a fair ground for comparison, it is essential to have a reference point. In order to provide a generic method for evaluating the complexity and hence enable comparison between architectures based on their associated parameter space, it is essential for the extraction to rely on normalized procedures, which can be realized by using standardized data models.

Various standards exist for modeling network state and configurations (*e.g.,* Structure of Management Information, Managed Object Format, *etc.*). In this work, we focus on the YANG data modeling language [9]. YANG models build upon a recognized standard in the industry, with major vendors supporting YANG releases of their implementations. They are also used by several standard bodies, *i.e.,* IETF, IEEE, ETSI. By design, the description providing by YANG models reflects the specifics of an implementation. In the last few years, the OpenConfig organization has been working towards the development of a set of vendor-neutral YANG data models based on a generic abstraction of networking elements (functions, services and protocols) [11]. Today's major vendors offer an OpenConfig-integrated version of YANG models of their implementations (see [19]). The availability of such a rich source of data, consolidated around a generic abstraction of networking functionality, makes YANG the ideal candidate for developing a methodology to determine and analyze the space of parameters associated with various network architectures.

# 3 OPLEX FRAMEWORK

The objective of OPLEX is to determine the space of state and configuration parameters associated with a network architecture by analyzing relevant YANG [9] data models. YANG [9] is a standardized data modelling language for network management protocols. It provides modeling primitives for network device state (read only parameter such as packet counter), device configurations (read/write parameters such as interface name, addresses, *etc.*), remote procedure calls and notifications. YANG organizes data definitions into hierarchies of schema nodes, *i.e.,* data structures of parameter definitions and attributes, grouped into modules. Each module constitutes a self-contained object that can be compiled.

## 3.1 Methodology

To determine the set of state and configuration parameters relevant to a network architecture, OPLEX analyzes YANG models by proceeding at three levels of representation: 1) at the module level by extracting state and configuration data; 2) at the device level by selecting modules corresponding to the functions used to achieve network operations, *e.g.,* switching protocol, redundancy mechanism, link aggregation feature, and 3) at the network instance level by analyzing the characteristics of the instantiation of the network architecture (*e.g.,* connectivity, activated interfaces). The extracted information is used to compute the dimension of the parameter space of a network architecture implementation

**Module analysis** Information about state and configuration data is contained in two types of nodes: `leaf` and `leaf-list`. `leaf` nodes are representations of state / configuration parameters which they model through an identifier and a data type; for instance `leaf interface_name type string`. `leaf-list` nodes are sequences of leaf nodes of a particular type. In a similar fashion to `leaf`, they come with an identifier, *e.g.,* `leaf-list vlan-id type string`. YANG enables multiple instances of `leaf` and `leaf-list` to be declared by defining them as child nodes of specific constructs called `list` nodes. `list` constructs are used to define an interior data node in the hierarchy of schema nodes. Each `list` can be the child node of another `list` node, forming as such dependency structures (example: `list destination-group (list destination (list config))`). Extracting these dependencies is essential as they contribute to the dimension of the parameter space. OPLEX determines the set of all `leaf` and `leaf-list` defined in a YANG module, as well as the `list` dependencies if relevant.

**Device analysis** In an operational context a subset only of the functions implemented in a device is actively employed. Examples of functions include for instance the type of protocol used to route traffic, the link aggregation feature selected to increase link capacity or the type of mechanisms triggered to provide redundancy guarantees. OPLEX determines all YANG modules relevant to active device-level elements of a specific network architecture implementation. Selecting the set of appropriate modules is however challenging given that parameters associated with an active function can be defined in more than one module. In this paper we achieve the selection through lexicographic matching between the conventional name of the protocols / technologies related to activated functions and the name of the YANG modules.

**Network instance analysis** `leaf`, `leaf-list` and `list` extracted from individual YANG modules define the network parameter space of a specific implementation. While the dimension of that space does not depend on the actual value of these parameters, it is affected by the size of `leaf-list` and `list`, which is driven by the instantiation of that specific implementation (for instance the number of configured interfaces depends on the number of devices and their connectivity). OPLEX consolidates the set of state and configuration parameters by extracting from network instance characteristics `leaf-list` and `list` size information.

**Parameter Space Dimension** We define the dimension of the parameter space of a network architecture implementation as a count on the total number of parameters that can be accessed via `read` and `write` operations. This depends on the number of YANG modules relevant to the functions activated on a device, the size of the sets `leaf` and `leaf-list` extracted from each module, and the size of the set of lists in the list dependencies associated with `leaf` and `leaf-list`, respectively. The dimension can be determined in a flexible way and be computed at the module level. It can also be agnostic to the specifics of either or both environment and activated functions. In this case, all set sizes take a default value of 1 (environment-agnostic) and all modules defined in the device are taken into account by default (function-agnostic). The formalization of the calculation of the dimension of the parameter space in described in detail in Appendix A.

## 3.2 Implementation

We implement the functionality of the OPLEX framework as part of a tool that can be used to automatically determine the dimension of the parameter space at the device level (as per the definition in Section 3.1) of any input network architecture. Our tool is designed to be operator-friendly: *i)* it is generic to any implementation for which YANG data models are available; *ii)* it is easily extensible, *i.e.,*

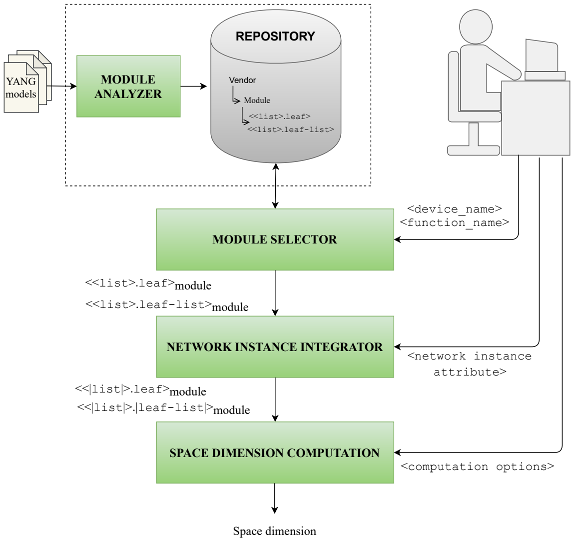

**Figure 2: Overview of the OPLEX tool.**

the repository of module data can be enriched at any time as the YANG models of different implementations become available; *iii)* it is practical, *i.e.,* the user can specify vendor, function and network characteristics information and select the options to determine the dimension, *i.e.,* at the device or module level, or if it is agnostic. The main components of the OPLEX tool are depicted in Fig. 2. It includes four main functions as follows:

**Module analyzer:** extracts all `leaf`, `leaf-list` and `list` parameters from an input set of YANG modules. Parameters are stored in a global repository, organized per vendor and per module.

**Module selector:** selects in the repository the set of `leaf`, `leaf-list` and associated `list` of all active functions related to a network architecture. Information about the activated functions and the device model (*i.e.,* vendor name) are provided as input either directly by the user, *e.g.,* the operator, or by interfacing the tool to an external (management) system.

**Network instance integrator:** determines `leaf-list` and `list` size information of the parameters associated with the set of modules selected by the module selector based on network instance characteristics provided as input in terms of attributes.

**Space dimension computation:** implements the functions described in Section 3.1 and Appendix A, and computes the relevant dimension(s) based on inputs received from the network instance integrator. Computation options (*i.e.,* device / module level, agnostic) can also be specified by the user. By default, the score is calculated for an OpenConfig device with all functions activated.

## 4 IXP USE CASE

IXPs constitute the core public infrastructure of the Internet. To be called IXP, at least three Internet operators need to be connected through the same peering Local Area Network (LAN). Internet Service Providers and Content Delivery Networks exchange Internet traffic through the IXP. IXPs not only reduce the portion of the traffic that an ISP delivered via its upstream transit providers, they also reduce latency and increase security.

We focus on the use case of IXPs for three main reasons. Their architecture and operations are well documented. Information is available from a rich set of publicly available data sources, which enables the evaluation of the impact of architecture options based on real data. In addition, given that all IXPs share a common goal and core service, it is possible to compare network architectures based on the knowledge required to operate alternative implementation options. Finally, we take advantage of our long-term involvement within the IXP community to collect a dataset of qualitative results regarding the perception of IXP operators with respect to their experience operating different types of architectures.

### 4.1 IXP Architectures and Parameter Space

To deliver connectivity services, IXPs can use different architectures, usually as a function of the size of their infrastructure. We investigate typical options deployed by IXPs today based on the document released by EURO-IX [30] that provides the list of features and architectures expected from network vendors. EURO-IX is the most significant association of IXPs, involving the largest players, *i.e.,* DE-CIX, AMS-IX, and LINX. To consolidate the list of common architectures, we also directly consult IXPs website and relevant press-releases (in particular, given that the wish-list was published in 2013[1]). We finally validated the extracted list with three vendors that provide IXPs.

The main architectures and protocols are shown on the first two rows of Table 1. Although the main service of an IXP is to provide layer-2 connectivity to Internet operators, it can be noted that in order to scale, their service can be implemented on a layer-3 overlay architecture. In general Link Aggregation (LAG) and Spanning Tree protocol implementations are used by small and medium IXPs, *i.e.,* with less than four switches, while larger IXPs rely on layer-3 VxLAN or layer-3 overlay type of networks.

We use our OPLEX tool to extract the parameter space of each of the architectures presented in Table 1 and determine its dimension based on an agnostic setting (see Section 3.1) that we denote $\delta_{agnostic}$. The results are reported in the last two rows of the table. The row before the last indicates the value of $\delta_{agnostic}$ at the device-level for the associated architecture. The last row presents the OPLEX score that we define as the ratio between the value $\delta_{agnostic}$ of the relevant architecture to the value $\delta_{agnostic}$ of the most straightforward layer-2 Link Aggregation (LAG) architecture (complexity score $2,386$). The highest relative complexity score (1.79) is obtained for the most complex layer-3 overlay architecture implementing ISIS, MPLS and BGP protocols.

### 4.2 Operator Survey and Consultations

To understand how operators themselves perceive the efforts they need to provide to manage various types of architectures, we conducted in 2020 a survey, and further direct consultations, among members of the IXP community. Eighteen IXPs and three leading vendors of the IXP market participated in the survey. Fig. 3 shows the geographical origin of the respondents, as well as their technical staff size that gives an idea of the involved *human resources*, and hence some qualitative insights onto the required management efforts. Most respondents are from Europe, which is consistent with

---

[1] It is worth highlighting that architecture changes are not rapid processes for IXPs.

| Architecture | Layer2 Only | | Layer3 VxLAN | | | Layer3 Overlay | | | |
|---|---|---|---|---|---|---|---|---|---|
| Protocols | LAG | STP | Static | IS-IS | OSPF | ISIS | OSPF | ISIS-BGP | OSPF-BGP |
| OpenConfig-aft | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| OpenConfig-bfd | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| OpenConfig-bgp | | | | | | | | ■ | ■ |
| OpenConfig-interfaces | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| OpenConfig-isis | | | | ■ | | ■ | | ■ | |
| OpenConfig-lacp | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| OpenConfig-local-routing | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| OpenConfig-mpls | | | | | | ■ | ■ | ■ | ■ |
| OpenConfig-network-instance | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| OpenConfig-ospf | | | | | ■ | | ■ | | ■ |
| OpenConfig-platform | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| OpenConfig-routing-policy | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| OpenConfig-stp | | ■ | | | | | | | |
| OpenConfig-system | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| OpenConfig-terminal-device | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| OpenConfig-vlan | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ |
| $\delta_{agnostic}$ | 2386 | 2504 | 2684 | 3202 | 2912 | 3499 | 3209 | 4274 | 3984 |
| **OPLEX score** | 1.00 | 1.05 | 1.12 | 1.34 | 1.22 | 1.47 | 1.35 | 1.79 | 1.67 |

**Table 1: IXP architectures and network parameter space dimension.**

the European concentration of IXPs [22]. While the panel misses representatives of some regions (*e.g.,* Latin America), it provides a ground for getting an initial perspective from members of the community.

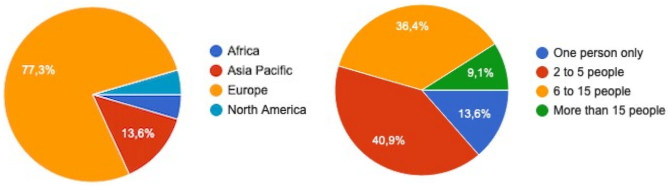

**Figure 3: Survey participants geographical origin and size**

We asked two main questions in the survey: 1) which level of certification is estimated to be necessary to operate different types of architectures, and 2) what perception of complexity - on a scale from 1 to 10 - the operator has with respect to these different architectures. The first question is used to understand the experience and knowledge required to operate and design a network, while the second is intended to put the notion of complexity into perspective from the point of view of the operator.

We used the networking industry training programs validated through certification exams as a knowledge scale for the first question. Certifications include four levels: entry, associate, specialist, and professional. The associated exams are all based on multiple-choice questionnaires and are usually taking place in a certification exam center. For the expert level, the exam also consists in configuring and implementing in a limited-time, state-of-the-art architectures on actual equipment at vendors’ certification centers.

Fig. 4 presents the responses to the first question in the form of a heatmap. It shows the distribution of the views of the respondents (color shade scale indicating the number of responses) to the combination Type of Architectures to Operate (x-axis) - Level of Expertise Required (y-axis). Most respondents estimate that an associate level is expected to operate L2-LAG. For other architectures, the results show that the expectations of respondents in terms of certification is more diverse, except for L3-Overlay architectures for which most respondents indicate that a professional-level is needed.

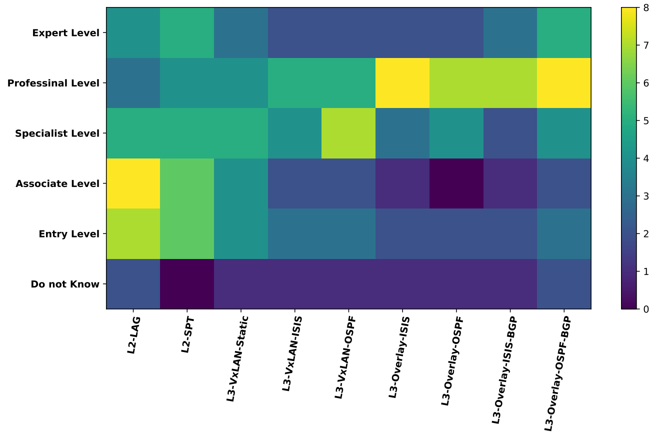

**Figure 4: Vendor certification evaluation**

The responses to the second question are shown as box plots in Fig. 5. The value 0 corresponds to the answer “I don’t know". The L2-LAG architecture is rated on average at 2. L3-Overlay architectures are usually rated as being two to three times more complex. To put the values reported in Table 1 in perspective with the perception of complexity as reported by IXP operators who responded to our survey, we superpose the two sets of results in a double y-axis figure in Fig. 6, with OPLEX scores on the left and operator complexity perception on the right. We can see that the size of the parameter space associated with an architecture and the perception of complexity in operating that architecture both increases in the correlated fashion.

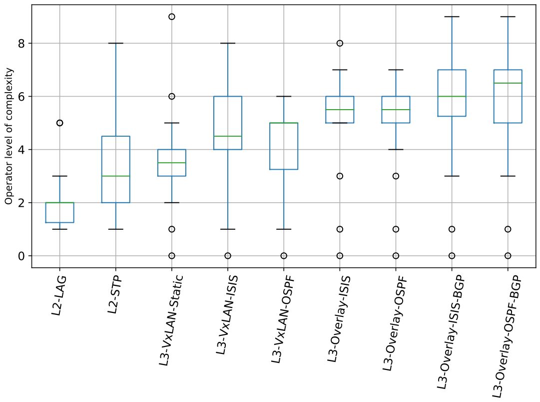

**Figure 5: Complexity level perception per architecture.**

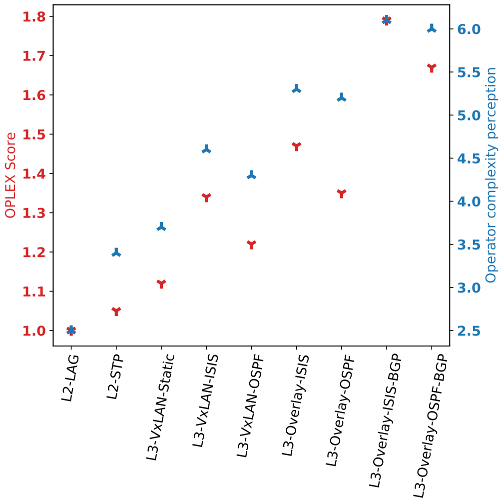

**Figure 6: OPLEX score *vs.* operator perception**

## 5 RELATED WORK

The work comes within the literature that has been focusing on developing methods for measuring the management and operational complexity of networked communication systems. Relevant approaches include the work by Brown *et al.* [17][26], with subsequent contributions by Clemm [18], that argue in favor of the definition of operator-facing metrics to quantify the complexity associated with managing network infrastructures. It also encompasses the work by Schönwälder [27] that proposes metrics to analyze the characteristics of Management Information Base (MIB) modules and that evaluates the usage of different features of the data models used in these modules MIBs. In addition, it includes the efforts initiated by Ratnasamy in [28] and further extended by Chun *et al.* in [6] that focus on the development of a conceptual framework for measuring the complexity of routing protocol implementations. Finally, it covers the proposals presented by Benson *et al.* in [3] and by Sun *et al.* in [8] and [4] which both depend on the analysis of network configuration files to determine a measure of operational network complexity. Our work is also motivated from developments in the software engineering domain where solutions for evaluating and managing configuration complexity span theoretical frameworks *e.g.,* [23], practical measurement tools *e.g.,* [24] and qualitative methodologies and best practices *e.g.,* [25]. By design, OPLEX builds on top of all these proposed solutions to evaluate the complexity of operating a network from the perspective of its configurations. It does however address important limitations of previous work by providing a solution that is easy for operators to use, independent of the specifics of internal configuration standards, and adapted to any type of network architectures for which YANG data models are available. In that respect OPLEX contributes to the efforts engaged in the recent years by White et al. [5][1] towards formalizing the concept of complexity for the design, deployment, maintenance and management of communication networks and computer networked systems.

## 6 DISCUSSION AND CONCLUDING REMARKS

IXPs, like the rest of the Internet, encounter massive traffic growth. To sustain increasing traffic volume, large IXPs tend to move to layer-3 overlay architectures, which comes with increasing complexity from the perspective of operating the network as illustrated by the obtained results in Section 4. Increasing traffic volume also means more devices to manage, which also contribute to increasing the complexity. In order to scale, IXPs face multiple complexity dimensions. OPLEX can help with automated complexity evaluation tools combined at the design phase and network management automation tools to identify the more suitable architecture and protocols stack to keep operations and management as simple as possible.

More generally, the OPLEX framework can be integrated as part of a general methodology to measure the "complexity" of operating a network. In particular, OPLEX enables the identification of the parameters that are essential to the network operations. In that respect, OPLEX is a step towards automatically extracting the set of parameters that are the most *semantically* significant in order to minimize operational efforts.

The development of OPLEX comes with a number of considerations. Being able to determine the exact value of the dimension of the parameter space is not discriminating when assessing complexity through the network factor. Given that the objective of OPLEX is to compare network architectures, what it is essential is to obtain comparative values, which can be achieved by using a reference point in terms of device model. However, while OPLEX focuses on the count of parameters at the device level, it does not take into account the relationship that exists between devices in a network, which also contributes to the complexity an operator faces when operating the network.

In the future, we will investigate how to integrate OPLEX with the model of device interactions proposed in previous work, *e.g.,* [6]. We also plan to extend OPLEX with more vendors and more functions, and take into account additional network characteristics, in order to evaluate other types of architectures.

## REFERENCES

[1] R. White, and J. E. Tantsura, "Navigating Network Complexity: Next-generation routing with SDN, service virtualization, and service chaining," Addison-Wesley Professional, 2015.

[2] M. H. Behringer, "Classifying network complexity," in *proc. of the Workshop on Re-architecting the Internet*, pp. 13-18, 2009.

[3] T. Benson, A. Akella, and D. A. Maltz, "Unraveling the Complexity of Network Management," in *proc. of NSDI*, pp. 335-348, 2009.

[4] X. Sun, and G. G. Xie, "Minimizing network complexity through integrated top-down design," in *proc of ACM CoNEXT*, pp. 259-270, 2013.

[5] M. Behringer, A. Retana, R. White and G. Huston, "A Framework for Defining Network Complexity," RFC 7980, 2016.

[6] B. G. Chun, S. Ratnasamy, and E. Kohler, "NetComplex: A Complexity Metric for Networked System Designs," in *proc. of NSDI*, vol. 8, pp. 393-406, 2008.

[7] M. Bruyere, C. Visser, D. Tuncer, "Toward Evaluating the Complexity to Operate a Network," *arXiv, 2201.03770, cs.NI*, January 2022.

[8] X. Sun, S. G. Rao, and G. G. Xie, "Modeling complexity of enterprise routing design," in *proc. of ACM CoNEXT*, pp. 85-96, 2012.

[9] M. Bjorklund, "The YANG 1.1 Data Modeling Language," RFC 7950, 2016.

[10] The YANG catalogue, https://yangcatalog.org/ - Accessed 06-06-23.

[11] OpenConfig Data Models, https://www.openconfig.net/projects/models/ - Accessed 06-06-23.

[12] B. Ager, et al., "Anatomy of a large European IXP," in *proc. of ACM SIGCOMM*, pp. 163-174, 2012.

[13] M. Bruyere, et al., "Rethinking IXPs' architecture in the age of SDN," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 12, pp. 2667-2674, 2018.

[14] PeeringDB, https://www.peeringdb.com/ - Accessed 06-06-23.

[15] Packet Clearing House, https://www.pch.net/services/internet_exchange_points - Accessed 06-06-23.

[16] IXP Database, https://ixpdb.euro-ix.net/en/ - Accessed 06-06-21.

[17] A. B. Brown, A. Keller, and J. L. Hellerstein, "A model of configuration complexity and its application to a change management system," in *proc. of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2005.

[18] A. Clemm, "Network Management Fundamentals", *Chapter 12 - Management Metrics: Assessing Management Impact and Effectiveness*, Cisco Press, 2007.

[19] OpenConfig YANG model repository, https://github.com/openconfig/public/tree/master/release/models - Accessed 01-06-21.

[20] D. Bogdanovic, B. Claise, and C. Moberg, "YANG Module Classification," RFC 8199, 2017.

[21] M. Bjorklund, "YANG Tree Diagrams," RFC 8340, 2018.

[22] Internet Exchange Map, https://www.internetexchangemap.com/ - Accessed 01-06-21.

[23] D. Krob, "Modelling of complex software systems: A reasoned overview," in *proc. of the International Conference on Formal Techniques for Networked and Distributed Systems*, Springer, Berlin, Heidelberg, 2006.

[24] J. Meinicke, et al., "On essential configuration complexity: measuring interactions in highly-configurable systems," in *proc. of the 31st IEEE/ACM International Conference on Automated Software Engineering*, pp. 483-494, 2016.

[25] B. Beyer, et al., "The site reliability workbook: Practical ways to implement SRE," *O'Reilly Media, Inc.*, 2018.

[26] A. B. Brown, A. Keller, and J. L. Hellerstein, "A Model of Configuration Complexity and its Application to a Change Management System Aaron," *IEEE Transactions on Network and Service Management*, vol. 4, no. 1, pp. 13-27, 2007.

[27] J. Schönwälder, "Characterization of SNMP MIB modules," in *proc. of the 9th IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 615-628, 2005.

[28] S. Ratnasamy, "Capturing complexity in networked systems design: The case for improved metrics," in *proc. of HotNets*, 2006.

[29] Lodhi, A. and Larson, N. and Dhamdhere, A. and Dovrolis, C. and claffy, k., "Using PeeringDB to Understand the Peering Ecosystem," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 44, no. 2, pp. 21-27, 2014.

[30] EURO-IX Wish-List, https://www.euro-ix.net/media/filer_public/0a/5b/0a5b4a4e-e032-41f8-b0f7-43c1375c5442/ixp-wishlist.pdf - Accessed 06-06-21.

## A DIMENSION OF THE PARAMETER SPACE

Let $\mathcal{M}_d$ be the set of YANG modules relevant to the functions activated on a device $d$. In addition let $\mathcal{I}_m$ be the set of `leaf` and $\mathcal{J}_m$ the set of `leaf-list` extracted from module $m$. We denote as $|j|$ the number of elements in `leaf-list` $j \in \mathcal{J}$. We also denote as $\mathcal{L}_i$ and $\mathcal{L}_j$ the set of lists in the list dependencies associated with `leaf` $i \in \mathcal{I}$ and `leaf-list` $j \in \mathcal{J}$, respectively. Let $|l_i|$ be the number of elements in list $l_i \in \mathcal{L}_i$ and $|l_j|$ be the number of elements in list $l_j \in \mathcal{L}_j$. The dimension $\delta_d$ associated with device $d$ is equal to:

$$\delta_d = \sum_{m \in \mathcal{M}_d} \left( \sum_{i \in \mathcal{I}_m} u_i + \sum_{j \in \mathcal{J}_m} v_j \cdot |j| \right)$$

with $u_i$ a variable equal to 1 if $\mathcal{L}_i = \emptyset$ and to $\prod_{l_i \in \mathcal{L}_i} |l_i|$ otherwise, and $v_j$ a variable equal to 1 if $\mathcal{L}_j = \emptyset$ and to $\prod_{l_j \in \mathcal{L}_j} |l_j|$ otherwise.

The dimension can be determined in a flexible way and be computed at the module level:

$$\forall m \in \mathcal{M}, \quad \delta_m = \sum_{i \in \mathcal{I}_m} u_i + \sum_{j \in \mathcal{J}_m} v_j \cdot |j|$$

It can also be agnostic to the specifics of either or both environment and activated functions. In this case, all $|j|$, $|l_i|$ and $|l_i|$ take a default value of 1 (environment-agnostic) and $\mathcal{M}_d$ includes all modules defined in device $d$ (function-agnostic):

$$\delta_{agnostic} = \sum_{m \in \mathcal{M}_d} \left( I_m + J_m \right)$$

with $I_m$ the size of set $\mathcal{I}_m$ and $J_m$ the size of set $\mathcal{J}_m$.