JNSM manuscript No.
(will be inserted by the editor)

Providing Optical Network as a Service with Policy-based Transport SDN

Marcos Siqueira · Fabian van 't Hooft · Juliano Oliveira · Edmundo Madeira · Christian Esteve Rothenberg

Received: date / Accepted: date

Abstract This paper presents a novel policy-based mechanism to provide context-aware network-wide policies to Software Defined Networking (SDN) applications, implemented with a *policy flow* based on property graph models. The proposal has been validated in a Transport SDN Controller (T-SDNC), supporting optical network virtualization via slicing of physical resources such as nodes, links and wavelengths, through use case testbed demonstrations of policy enforcement for SDN applications, including optical equalization and virtual optical network control. Additionally, the policy engine incorporates a simulation-assisted pre-setting mechanism for local policy decisions in case of problems in communication with the controller.

Keywords Transport SDN · Optical Network Virtualization and Policy-based SDN

1 Introduction

The concept of network virtualization emerged as a proposal of extending the notion of virtualization of computational resources to that of network resources. Network virtualization can be defined [1] as *"a technique for isolating computational and network resources through virtualization to allocate them to a logical (virtual) network for accommodating multiple independent and programmable virtual networks"*. Differently from Virtual Private Networks (VPN) which provide connectivity isolated from other VPNs over a

Marcos Antonio de Siqueira, Edmundo R. M. Madeira and Christian Esteve Rothenberg
University of Campinas (Unicamp) - Campinas - SP, Brazil, CEP 13083-852
E-mail: marcos.siqueira@gmail.com, edmundo@ic.unicamp.br, chesteve@dca.fee.unicamp.br

Fabian N.C. van 't Hooft and Juliano Rodrigues F. de Oliveira
CPqD Telecom R&D Center - Campinas - SP, Brazil, CEP 13086-902
E-mail: fabian,jrfo@cpqd.com.br

shared network infrastructure, network virtualization aims to achieve additional features such as programmability (such as allowing independent control plane per virtual network), topology awareness, network abstraction, quick reconfiguration, and resource isolation. Notwithstanding, one of the major motivations of network virtualization is the provision of Network as a Service (NaaS) [2].

The implementation of new optical networking technologies and the need for providing dynamic services, including virtualization, impose many challenges for transport networks operation. Previously simple routines such as spectrum equalization, optical gain control, channel isolation, channel instantiation and even the Generalized Multi-Protocol Label Switching (GMPLS) control plane [3] have to be re-visited, in order to be adapted to this new scenario.

The road to SDN is being regarded as an architectural approach to separate network control planes from the data-forwarding plane, enabling virtualization capabilities (i.e., isolated and transparent resource sharing), automation of programmable network operations, and easier deployment of new services based on software implemented features defined in a logically centralized control plane.

In addition, much effort has been devoted on the research and implementation of Policy-Based Network Management (PBNM) [4], aiming at mapping of high level business policies to the appropriate configuration of network devices by the management systems, thus reducing the complexity of network operation, independent of type, model, vendor and operating system used. Undoubtedly, the concept of policy is central to SDN and recent efforts [5] have started to focus on policy languages and abstractions tailored for SDN.

This paper presents our ongoing work on a T-SDNC [6] that leverages a policy-based engine based in the concept of a *policy flow* [7]. Advancing our prior work, this paper introduces a context-aware policy-based control mechanism for SDN applications, which is able to govern the *i*) behaviour of SDN applications, *ii*) access control to network resources, and *iii*) concurrency model. The policy evaluation is performed using a graph-based approach, where graph information is automatically created based on models. We present two use cases of policy enforcement for SDN applications, namely, Optical Equalization SDN Application (EQUAL-APP) and Virtual Optical Network Configuration and Restoration SDN Application (VONCR-APP). The latter allows to dynamically instantiate a given Virtual Optical Network (VON) based on network entry points, wavelength demand matrix and protection requirements. Additionally, after some failure at the physical network, VONCR-APP is able to reconfigure the client VON to keep fulfilling the service requirements.

The next sections of this paper are organized as follows: Section 2 presents related and previous works, section 3 presents the proposed architecture and implementation, Section 4 presents simulated and experimental results and Section 5 concludes the paper with final remarks and presents some avenues for future work.

2 Related Work

Recently, the concept of NaaS is being proposed in the context of optical transport networks [2]. The driving applications are datacenter interconnection, optical infrastructure sharing among different service providers, and IP-optical integration. In this context, Pfeiffer et al. [8] present a model named "open access" which enables mutually independent utilization of the available optical resources by different client systems, potentially employing different modulation formats, bandwidths and multiplexing technologies, employing different topologies. On the other hand, Schoentgen et al. [9] study network sharing advantages and strategies in developing countries, including national optical fiber backbone for network sharing between telecom operators. Nevertheless, the research activities on NaaS and network sharing in optical networks and virtualization is in its early stage. For instance, Nejabati et al. [10] propose an "optical virtualization layer" which is responsible for creating and monitoring virtual optical networks, considering physical layer constraints and impairments.

In previous related work [11], we have proposed hybrid SDN-IP networks with the *RouteFlow* control platform, a commodity routing architecture that combines the line-rate performance of commercial hardware with the flexibility of open-source routing stacks running on general purpose hardware, allowing the implementation of IP-routed NaaS. In the context of optical transport networks, we have been working on enabling virtualization and autonomic operation by means of a SDN controller that implements an abstraction layer called Transport Network Operating System (T-NOS) providing network resource slicing [6]. Recently, we demonstrated the virtualization of the GMPLS protocol stack, running separate control plane instances for each network slice, as an application of the SDN controller [12]. Additionally, PBNM mechanisms have already been explored by the authors in [7] and [13].

Over the last decades, there have been plentiful research efforts [14] that shall be regarded as intellectual contributions that have lead to the current thinking on SDN. A clean separation of control and data planes is at the core of SDN –and it's the rule in optical networks. Network virtualization and SDN have a symbiotic relationship that is currently being explored by many in different network domains. Our focus is on virtualizing optical transport networks and easing the programmability of control applications following principles of SDN such as logical centralization of network-wide views and introduction of useful control abstractions. Recent related work include the research on new programming languages for SDN [5]. A noteworthy example is Pyretic [15], a language and runtime system for developing SDN applications with focus on packet forwarding policies. Pyretic introduces an abstract packet model, an algebra of high-level policies and network objects, including, for instance, a fabric policy which allows computing a graph algorithm on the underlying topology, as well as performing mapping, paving the way to deploy network virtualization solutions.

Notwithstanding, while the SDN concept is more extensive, including more functions than PBNM, in the proposed architecture the T-SDNC requires policies to govern its own operation. More recently, declarative policy architectures have been proposed in the context of SDN [16]. Altogether, the issue of policy management and enforcement in virtualized infrastructures is an ongoing research thread [17].

3 Proposed Architecture and Implementation

3.1 Transport SDN Controller

The system architecture depicted in Figure 1, firstly introduced in [6], is comprised by a set of Optical Network Elements (O-NE) and the T-SDNC. The O-NE runs a node controller daemon for node abstraction and control via Network Configuration Protocol (NETCONF), and Link Layer Discovery Protocol (LLDP) for neighbor discovery, allowing automatic network bootstrapping. The T-SDNC aims to allow autonomic control of the optical network, network slicing, supporting SDN applications. The T-SDNC provides a network functions API for plugging control applications to perform specific tasks, taking advantage of the network abstraction. The targets of our SDN applications include cognitive and adaptive controls for autonomic optical adjustments such as global equalization, global Erbium Doped Fibre Amplifier (EDFA) gain control, Dense Wavelength Division Multiplexing (DWDM) system auto alignment, fault prediction and preventive actions. Additionally, it provides an application with a virtualized GMPLS control plane, allowing simplification and offloading the O-NE, while maintaining compatibility with legacy transport networks.

3.2 T-SDN Applications

One of the purposes of this work is to define and demonstrate enforcement of policies to SDN applications. Therefore, the following Transport SDN (T-SDN) applications have been considered as use cases for validation purposes:

- **Spectrum equalization and optical gain control**: these mechanisms can be performed in many different ways. Commonly, they are performed by channel attenuators of Wavelength Selective Switch (WSS), and by optical amplifiers gain control. In reconfigurable networks, some of the optical equalization mechanism should be coordinated by a T-SDN application in order to achieve better performance than traditional mechanisms (GMPLS-based) by using global information about the network, as described in our previous work [18], while avoiding overshoots and undershoots. Additionally, the Network Elements should support uncoordinated equalization mechanisms in order to allow the network to operate sub-optimally in case of communication failure with the T-SDNC.

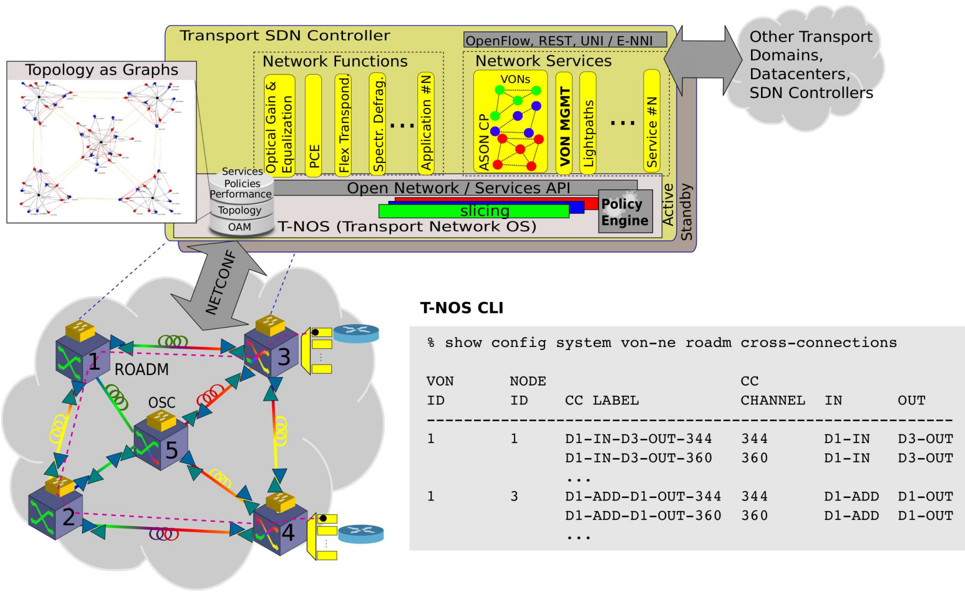

Fig. 1 System Architecture

– **Protection and restoration**: optical transport networks have strong requirements for services availability. Therefore, different levels of lightpath protection and restoration are employed, such as re-routing in case of failures. Therefore, protection and restoration policies can be defined in the T-SDN controller at lightpath granularity in order to enforce protection and restoration in the case of failures such as fiber cut, or equipment failure. In the case of providing transport network virtualization services, such as NaaS, besides restoring lightpaths, the network should be able to restore VON topologies and services in the case of failures in the physical infrastructure.

3.3 Graph-based Policy Architecture

Most of the policy architectures proposed in literature are built over a policy information model and a policy engine. Policies are commonly defined as *events*⇒*conditions*⇒*actions* with the main goal of automatically mapping high level policies to network specific configurations based on an imperative mechanism, where the policy manager is responsible for enforcing the policies via distributed Policy Enforcement Point (PEP).

In a previous work [13], the concept of *policy flow* has been introduced, where policy conditions and actions are evaluated in a distributed pipeline, allowing a distributed policy engine, via an approach similar to mobile agents. Additionally, the policy evaluation includes the concept of "situation" identification, allowing a more broad analysis than just evaluating a set of conditions. The referred architecture allows implementing a declarative policy system, since decisions are distributed, based in local state and input parameters transported by the policy flow, which can be propagated with the promise of each node that it will do a set of internal configurations in order to achieve the desired state. This approach is in line with recent declarative approaches for policy specification in SDN [16] and virtualized environments [17].

In this work, we propose a new approach for distributed policy systems, using the concepts of graph modelling, analysis and traversal. As the network components, its correlations and constraints are modelled in NETCONF Data Modeling Language (YANG), and through transformations, a graph database can be created. By employing graph-based algorithms at the component level, powerful abstractions can be created, hiding the complexity of subsystems and allowing these to be easily mapped to the policy engine. As the T-NOS and the policy model are based in configurations modelled in YANG with transformations for representing the network, internal node architecture and policies as graphs, the configuration of network infrastructure, virtualization substrate and policies can be performed, for instance, via NETCONF. Nevertheless, while the network operator has full visibility of the managed objects, each client has access only to its own network slice.

With a centralized graph database, policies can be modelled in novel ways by means of property graph operations, such as graph traversal, sub-graph, listenable graphs, edge and vertex attribute analysis, and others. In order to allow policy engine distribution, as supported in [13], the concept of graph composition [19] can be used, via distributed graph databases[1] or via replication of abstracted sub-graphs at the network operating system at the T-SDNC. This model presents benefits of allowing extending the graphs with "aspects" or "namespaces" incorporating additional information natively in the data model via templates or instantiators, without having to roll-out new software.

The proposed *policy flow* (*event* ⇒ *situation identification* ⇒ *condition analysis* ⇒ *actions selection and execution*) is depicted in Figure 2. After some failure in a receiver, such as an optical transponder, the situation identification is performed over current network state information using graph analysis via graph database queries (shown in pseudo-code queries, similar to the Cypher[2] language). The situation analysis allows the verification of potential interfaces that could be re-configured in order to solve the problem and if additional channels have been affected. After that, pre-configured system policies are evaluated in order to determine if and which actions should be performed. After action execution, situation is re-checked in order to determine if the actions performed have solved the problem, and if not, other actions could be triggered. For instance, if the channel high Bit Error Rate (BER) could not be solved via optical equalization, and if the channel if critical (business requirement), and other conditions are satisfied, the lightpath could be re-

[1] http://thinkaurelius.github.io/titan/
[2] http://www.neo4j.org/learn/cypher

routed. Note that different T-SDN applications may be called within the *policy flow*.

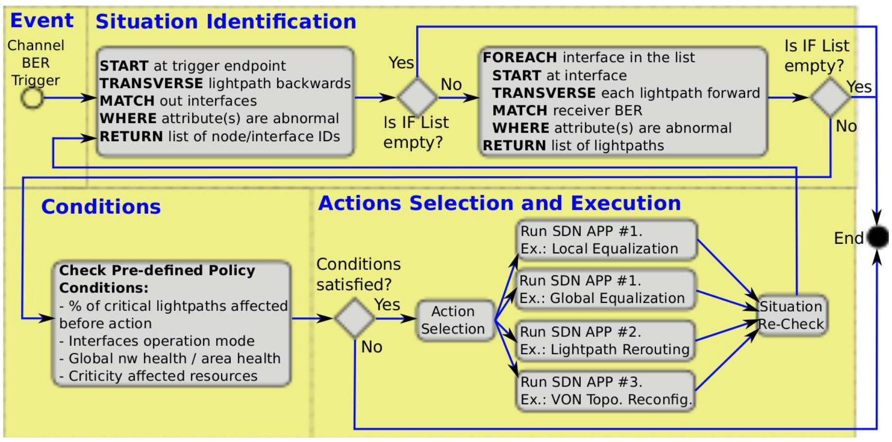

Fig. 2 Example of Equalization Policy Model

In the scope of this work, the main goal of the policy engine is to provide policy-based control of the network infrastructure and the virtualization substrate. The proposed architecture allows the VON-provider to expose an abstracted topology to the client, mapped by the T-NOS to several physical nodes interconnected in a mesh topology. In this sense, each client is responsible for managing its own VON resources and policies by means of its own mechanisms, in a similar way as if they had their own physical network to manage.

4 Use Cases and Experimental Results

4.1 Dynamic VON Reconfiguration

Our first use case related to policy evaluation is the SDN application called VONCR-APP, designed to perform instantiation of virtual optical networks and their reconfiguration after link or node failure in the physical network. VONCR-APP instantiates VONs based on input parameters such as: client entry points, survivability- and bandwidth-requirements, level of network elements abstraction desired, requirement of a virtualized control plane by the client, and others. By using the network graph and the Path Computation Element (PCE), a set of path computations are performed for working-, protection- and restoration- lightpaths. Using this information a minimum amount of resources is allocated for the given VON. When the VON graph is defined, the network resource slicing is performed in the T-NOS via XML transformations, virtualized GMPLS control plane is instantiated for the given VON, if needed, and the specific VON policies are configured.

The experimental network, shown in Figure 3c, is composed of five DWDM network elements and the T-SDNC. Each node includes a multi-degree flexgrid Reconfigurable Optical Add-Drop Multiplexer (ROADM), optical amplifiers, supervisory channels and optical transponders. The T-SDNC, running the T-NOS and virtualized eXtensible Control Plane (XCP) instances is equipped with a Intel Core i7, 2.8GHz CPU, with 8GB of RAM. Topology discovery is performed by means of LLDP which runs over the supervisory channel of the nodes. Topology information from all the nodes is consolidated automatically at the T-NOS, and used by the XCP manager daemon in order to instantiate the “Main VON”, containing all ROADM nodes and DWDM links. Therefore, one Linux Containers (LXC) virtual machine is instantiated for each real node. Additionally, the XCP manager starts the GMPLS stack in each Virtual Machine (VM), and creates TE-Links connecting the control planes, according to the physical topology. VONCR-APP will allow additional VON to be instantiated automatically, based on an XML input describing the VON entry points, demand matrix and redundancy requirements. Figure 3a shows two VON: green and blue, with their mapping to the physical network and their respective TE-links auto-configured by the XCP manager.

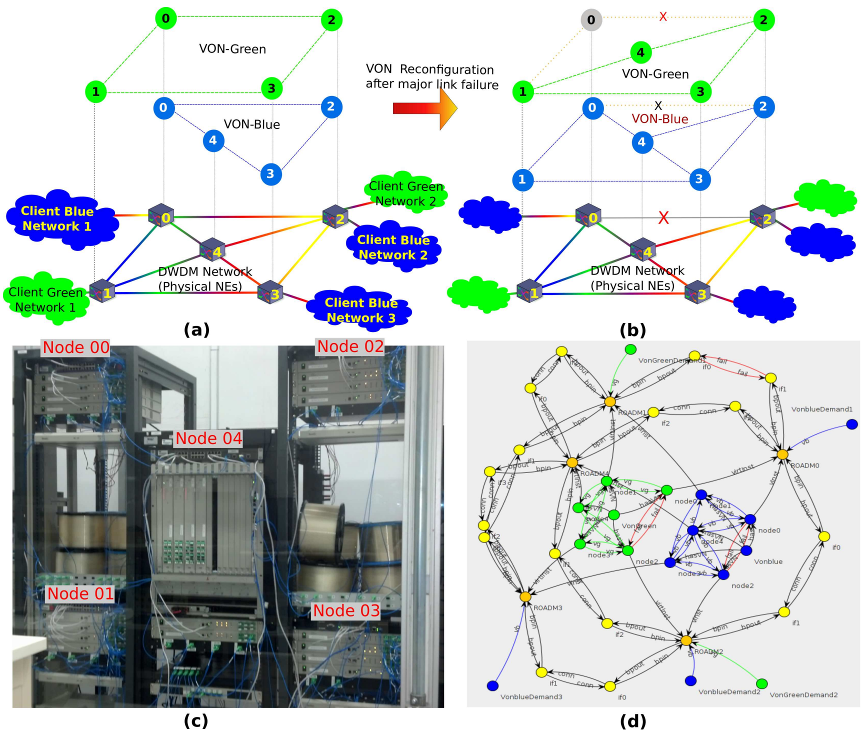

Fig. 3 VON reconfiguration after major failure

Figure 3b shows the expected reconfiguration for the VON after a major failure in the link connecting Nodes 1 and 3. The steps are described above:

- A failure, such as optical fiber cut, occurs in the link connecting Nodes 1 and 3.
- If client networks are provisioned with protection, it switches traffic to the alternate route.
- The failure is informed to the T-SDNC via NETCONF notification, and synchronizes the topology graph attributes.
- An event is triggered for the policy engine, which, in turn, traverses the graph, analysing the network situation and high level policies. In this case, one of the policies states that a given VON MUST provide at least two alternate paths to each pair of demands.
- After graphs traversals, the selected policy actions are to call VONCR-APP to reconfigure both VON as depicted in Figure 3b.
- The application VONCR-APP adds new resources to each VON at the T-NOS, in order to maintain the requirement of disjoint alternate route for every pair of client networks. Additionally, new XCP instances and TE-Links are deployed if needed.
- It's up to the client of each VON to manage lightpath re-routing via the new path made available by VONCR-APP.

We have implemented a first prototype of the situation analysis using the Gremlin[3] graph language, including graph traversals and attributes verifications. For that, an abstracted graph of the network has been generated, where each ROADM is abstracted as a general switching fabric and a set of interfaces. Therefore, all the complexity of each component such as WSS, Optical amplifiers, multicast switch, is standardized, in a way that the policy model could be reused when node architecture changes. Figure 3d depicts the abstracted graph visualization, including both VON after the actions performed for VON reconfiguration. JUNG[4] Java API has been used to generate the graph visualization.

4.2 Optical Equalization Local Policy

Our second use case addresses a common yet challenging requirement of optical transport networks that need to continue operating without interruption despite loss of connectivity with the management plane –and even when there are failures in the control plane. Therefore, T-SDN architecture has to fulfil this requirement, and functionalities which are essential for network operation have to be performed in a distributed way by network equipments.

Some of these functionalities could be implemented by both the T-SDNC and the network elements. The T-SDNC could have a richer set of features, taking advantage of computing power and global network information provided

[3] https://github.com/tinkerpop/gremlin/wiki
[4] http://jung.sourceforge.net/

by the graph abstractions, and the network element could have just a subset of the functionalities, enough to maintain the network operating in the case of failure in the communications between the network element and the T-SDNC.

In order to demonstrate this concept, we have implemented a mechanism for optical equalization, which is able to perform optimal global equalization when running as an application of the T-SDNC, as detailed in [18], or locally by the ROADM in the case of a T-SDN control plane failure. The mechanism has been designed in a way that the ROADMs are able to perform equalization in a sub-optimal manner, based on previous instructions sent by the T-SDNC. These instructions are created based on simulations, performed by a specific application at the T-SDNC, named *Equalization Simulator*. The simulator gathers specific information about the target network, such as node sequence for each channel (lightpaths), input power variation along time, requirements on maximum overshoot and undershoot as well as convergence time (based on pre-defined policies). Using these variables, the *Equalization Simulator* evaluates failure conditions, such as fiber attenuation, at different spans. Based on the requirements and simulation results, the T-SDNC pre-sets the equalization mechanism to be employed by each ROADM in this case.

Figure 4a shows a simulation model created using Matlab Simulink, where the equalization mechanism has been implemented as discrete events. Each ROADM has an independent mechanism, receiving as input signal the output of the previous ROADM. The model considers that the value received by each block is the optical power (in dBm) of one channel. The model can be created automatically, based on the network topology discovered by the T-SDNC. On the other hand, requirements, such as overshoot and convergence time have to be pre-defined as system policies.

The equalization mechanism of each ROADM is implemented as a Proportional, Integral, Derivative (PID) controller. Additionally, validation of the stability of the measurement can be performed checking the standard deviation (σ) of the last power values measured. In this case, the equalization mechanism only acts if the signal is stable for a given amount of time. For all the simulations, the input power at the first ROADM is 11dBm, the target power at the output of each ROADM is 4dBm. Span- and ROADM-losses and optical amplifiers are not modelled, since it is assumed that the amplifiers are working in Automatic Gain Control (AGC) mode, maintaining fixed gain thus compensating losses. At time=20s a power degradation of 10dB is simulated at the input of the system, and for the last four graphs other degradations of 5dB are simulated in the last span. The graphs shown in Figure 4, are results of simulations used to pre-set equalization mechanisms at the ROADMs:

- **Simultaneous actuation:** Figure 4b depicts output power of each span versus time, with an input signal varying up to $0.4dB$. Using $Kp = 0.6$, no σ validation, the unordered execution of the equalization mechanisms of the different ROADMs caused an overshoot of approximately $6dB$ and needed more than 12 seconds to converge after the failure.

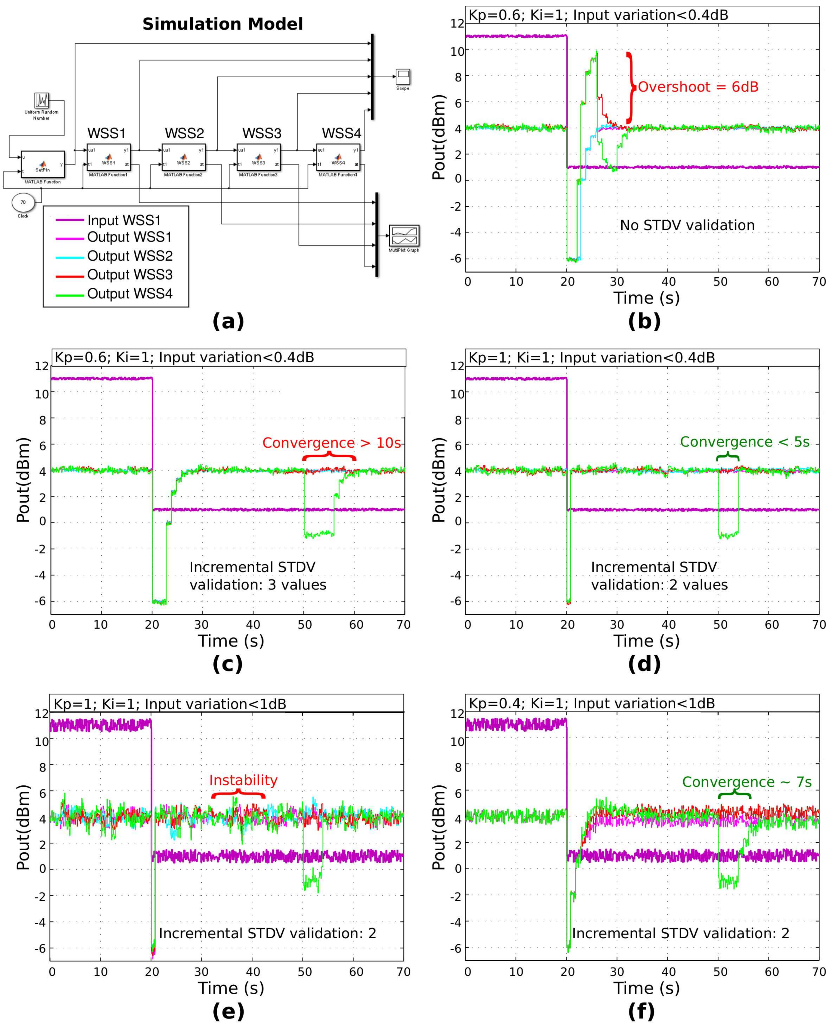

Fig. 4 Equalization Simulation Results

– **Sequenced actuation:** the results shown in Figure 4c were obtained by employing a sequencing mechanism developed as one of the contributions of this paper. The mechanism consists of configuring the stability condition validation, where the ROADM number N uses the current measurement, N and $N+1$ to verify signal stability. This simple mechanism guarantees that the second ROADM in the chain only starts equalization when the first finishes, and so on, without signaling or coordinated control requirements. Nevertheless, the picture shows that, besides having zero overshoot, when the degradation occurs in the last span, the convergence time is not acceptable, since the stability condition needs $N+1$ seconds to stabilize.

- **Accelerated sequenced actuation:** since in the last simulation the overshoot was zero and the problem was the convergence time, the controller may be more aggressive on stability validation and Kp. Therefore, with $Kp = 1$ and σ using only current value and the Nth measurement, where N is the ID of each node, the convergence for the worst case is 5 seconds, as depicted in Figure 4d. Therefore, in this scenario, these parameters can be used to pre-set the local equalization mechanisms of the ROADMs for the current network situation.
- **Input signal with high variation:** if the input signal varies with higher intensity ($1dB$ instead of $0.4dB$), the pre-setting from Figure 4d will lead to instability, as shown in Figure 4d. Therefore, the simulator would have to be consulted for another optimal pre-setting.
- **Stabilization with input variation:** in order to minimize instability with input variation, Kp was reduced to 0.4 and equalization threshold was increased to $1dB$. Even though the results depicted in Figure 4f show an increase in convergence time to 7 seconds, this setting provides for a more stable network.

5 Conclusion and Future Work

This paper proposed a novel way to control T-SDN applications employing Policy-Based Network Management concepts embodied as a *policy flow* which includes “network situation” evaluation, using graph database queries. The proposal extends our previous work [6] which introduced a SDN controller for transport network virtualization and autonomic operation, since the proposed policy engine is one of the building blocks of the T-SDNC.

In order to validate the proposal, a *policy flow* which deals with faults and performance issues has been proposed and prototyped using the Gremlin graph language. The *policy flow* allows situation evaluation and selection of the most adequate application to call, such as global equalization, optical gain control, or virtual optical network logical-to-physical mapping reconfiguration. Additionally, a simulation-assisted local policy mechanism has been proposed and demonstrated, which allows network elements to dynamically react to specific situations in case of failure in communications with the T-SDNC or in case of strict timing requirements. The concepts have been validated in a real testbed network comprised by five flexgrid multidegree WSS ROADMs.

As future work we intent to: (1) evolve our implementation, increasing the level of integration of the different building blocks; (2) perform scalability tests; (3) implement a distributed T-SDNC, using the concept of distributed graph databases; and (4) extend the concept and implementation to multilayer networks, including packet-based networks using OpenFlow programmability.

Acknowledgement

This work was partially supported by FUNTTEL, FINEP, and CPqD.

References

1. A. Nakao, “Network Virtualization as Foundation for Enabling new Network Architectures and Applications,” *IEICE transactions on communications*, vol. 93, no. 3, pp. 454–457, 2010.
2. “Is It Finally Time for Network as a Service?.” Ciena Whitepaper, 2013.
3. E. Mannie, “Generalized Multi-Protocol Label Switching (GMPLS) Architecture.” IETF RFC 3945 (Proposed Standard), Oct. 2004.
4. J. Strassner, *Policy-based Network Management: Solutions for the Next Generation*. Morgan Kaufmann, 2003.
5. N. Foster, A. Guha, M. Reitblatt, A. Story, M. Freedman, N. Katta, C. Monsanto, J. Reich, J. Rexford, C. Schlesinger, D. Walker, and R. Harrison, “Languages for Software-defined Networks,” *Communications Magazine, IEEE*, vol. 51, no. 2, pp. 128–134, 2013.
6. M. Siqueira, J. Oliveira, G. Curiel, A. Hirata, F. Hooft, M. Nascimento, C. E. Rothenberg, and J. Oliveira, “An Optical SDN Controller for Transport Network Virtualization and Autonomic Operation,” *Globecom 2013 Workshop of SDN in Optics*, 2013.
7. M. Siqueira, N. A. Nassif, R. A. Resende, and M. Lima-Marques, “Policy-Based Architecture for QoS Management in Enterprise IP Networks,” in *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management*, 2005.
8. T. Pfeiffer, “New Avenues of Revenues: Open Access and Infrastructure Virtualization,” in *National Fiber Optic Engineers Conference*, Optical Society of America, 2012.
9. A. Schoentgen, “Network Sharing: a Hot Topic for Operators and Regulators in Sub-Saharan Africa,” *Communications and Strategies*, no. 86, p. 43, 2012.
10. R. Nejabati, E. Escalona, S. Peng, and D. Simeonidou, “Optical Network Virtualization,” in *Optical Network Design and Modeling (ONDM), 2011 15th International Conference on*, pp. 1–5, IEEE, 2011.
11. C. E. Rothenberg, M. R. Nascimento, M. R. Salvador, C. N. A. Corrêa, S. Cunha de Lucena, and R. Raszuk, “Revisiting Routing Control Platforms with the Eyes and Muscles of Software-defined Networking,” in *Proceedings of the first workshop on Hot topics in software defined networks*, HotSDN ’12, (New York, NY, USA), pp. 13–18, ACM, 2012.
12. J. Oliveira, M. Siqueira, G. Curiel, A. Hirata, F. van t Hooft, D. Macedo, M. Colazza, and C. E. Rothenberg, “Experimental Testbed of Reconfigurable Flexgrid Optical Network with Virtualized GMPLS Control Plane and Autonomic Controls Towards SDN,” in *Proceedings of International Microwave and Optoeletronics Conference*, SBMO and IEEE, 2013.
13. M. A. Siqueira, F. L. Verdi, R. Pasquini, and M. F. Magalhaes, “An Architecture for Autonomic Management of Ambient Networks,” in *Autonomic Networking*, pp. 255–267, Springer, 2006.
14. N. Feamster, J. Rexford, and E. Zegura, “The Road to SDN: An Intellectual History of Programmable Networks,”
15. C. Monsanto, J. Reich, N. Foster, J. Rexford, and D. Walker, “Composing Software Defined Networks,” *NSDI, Apr*, 2013.
16. M. Smith, M. Dvorkin, Y. Laribi, V. Pandey, P. Garg, and N. Weidenbacher, “OpFlex Control Protocol.” Internet Draft, 2014.
17. P. Balland and T. Hinrichs, “Congress: a System for Declaring, Auditing, and Enforcing Policy in Heterogeneous Cloud Environments,” in *Openstack Summit Atlanta*, May 2014.
18. E. C. Magalhães, J. Oliveira, H. Carvalho, M. Magalhães, M. Garrich Alabarce, M. Siqueira, A. Bordonalli, and J. Oliveira, “Global ROADM-Based Spectrum Equalizer in SDN Architecture for QoT Optimization at DWDM Networks,” in *Optical Fiber Communication Conference*, pp. W2A–35, Optical Society of America, 2014.
19. M. Brocheler, A. Pugliese, and V. Subrahmanian, “COSI: Cloud Oriented Subgraph Identification in Massive Social Networks,” in *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, pp. 248–255, IEEE, 2010.

Biographies

Marcos Antonio de Siqueira is currently a Systems Engineer at Padtec Optical Systems. He received the B.S. degree in Electrical Engineering in 1999 from UFG, Brazil, the M.Sc. degree in Electrical Engineering in 2002 and is concluding the Ph.D., both at Unicamp, Brazil. His R&D activities are related to packet-optical networks, policy-based systems, network control plane, SDN, and others. He has over 15 publications in conferences and hold professional certifications such as CCNA, JNCIA, CCSE, PMP.

Fabian Nicolaas Christiaan van 't Hooft is an Optical System Engineer at CPqD and responsible for the NextGen-networks. He received his B.S. degree in Electrical Engineering in 2000 from HTS-Haarlem (Netherlands), Diplom-Informatiker degree in 2002 from the University of Osnabrueck (Germany) and is concluding his Ph.D. Thesis at the Institute of Computing at Unicamp, Brazil. Worked in Industry in Germany for the largest ISP in the fields of Billing, Data-analysis and network-management. Fabian has 7 publications in the field of network-management.

Juliano Rodrigues Fernandes de Oliveira received his Ph.D. at University of So Paulo (USP), M.Sc. degree at University of Campinas (UNICAMP) and B.S. degree from University of Campina Grande (UFCG), Brazil, all in Electrical Engineering. He currently leads CPqD Foundation optical systems and subsystems R&D, transferring 9 innovative equipments to the Brazilian industry. He has registered 6 patents and published up to 40 international conferences and journals papers.

Edmundo Roberto Mauro Madeira is a Full Professor at the Institute of Computing of University of Campinas (UNICAMP). He received his Ph.D. in Electrical Engineering from UNICAMP. He has published over 140 papers in national and international conferences and journals. He has also supervised more than 50 Master and Ph.D. students. Currently, he serves as editorial board member for the Springer JNSM. He was General Chair of the LANOMS 2011, and was Technical Program Co-Chair of the IEEE LatinCloud 2012. His research interests include network management, virtualization and cloud computing.

Christian Esteve Rothenberg is an Assistant Professor at University of Campinas (UNICAMP), where he received his Ph.D. in Electrical and Computer Engineering in 2010. From 2010 to 2013, he worked as Senior Research Scientist at CPqD R&D Center in Telecommunications, Brazil. He holds the Telecommunication Engineering degree from the Technical University of Madrid (ETSIT-UPM), Spain, and the M.Sc. degree from Darmstadt University of Technology (TUD), Germany. He has two international patents and over 50 publications in scientific journals and top-tier networking conferences such as SIGCOMM and INFOCOM.