Virtual Topology Reconfiguration in Optical Networks by Means of Cognition: Evaluation and Experimental Validation [Invited]

Natalia Fernández, Ramón J. Durán Barroso, Domenico Siracusa, Antonio Francescon, Ignacio de Miguel, Elio Salvadori, Juan Carlos Aguado, and Rubén M. Lorenzo

***Abstract*—In optical networking, virtual topologies have been introduced mainly to provide service providers with logical connections equipped with a reserved amount of bandwidth, which can be exploited to interconnect their equipment at the edges of the transport infrastructure. Virtual topologies are thus basically an abstraction of the real substrate, created by means of a process called virtual topology design (VTD). VTD is a complex task, affected by many parameters and constraints, and among them current traffic conditions are very relevant. Indeed, it is possible that after a certain time a virtual topology becomes inappropriate to serve current traffic. In such cases, the virtual topology can be reconfigured by creating new lightpaths or modifying or deleting existing ones, thus possibly creating some service interruptions. In this paper a new virtual topology reconfiguration technique is presented. In this technique, a cognitive entity designs and reconfigures virtual topologies by exploiting traffic forecasting solutions and taking advantage of past history. Moreover, a new transition method is also proposed to reduce the impact of instable routing tables during the reconfiguration process. We demonstrate, by means of simulation, the advantages of the proposed methods, as they reduce both the operational costs and the resources in operation while maintaining low packet loss ratio. Furthermore, we validate the operation of the proposed solutions in an emulated testbed.**

***Index Terms*—Cognitive techniques; Cost saving; Optical networks; Virtual topology reconfiguration.**

I. Introduction

Optical technologies are facing an unceasing evolution aimed at providing ever increasing capacity [1] and flexibility [2] to the owners of infrastructure. In parallel to that, network operators have been scouting new business opportunities to be captured by means of innovative services exploiting the enhanced capabilities provided at the physical layer. In the middle stands the control and management plane of the network, which is expected to translate the needs and requests of the new coming network applications into mechanisms and commands for the physical layer.

One of the most interesting networking applications is the abstraction of the infrastructure by means of a *virtual topology*. More specifically, here a virtual topology is meant to be the *set of* (potentially all) *the lightpaths in a network*. A service provider (or a large institution) can lease from the network operator owning the infrastructure the bandwidth (i.e., the optical channels) provided by such a virtual topology to provide connectivity between its end equipment (in the case of an IP/WDM network, these are the IP routers). This may hold for a single or multiple virtual topologies coexisting together and belonging to one or more service provides. From the control and management perspective, the owner of the infrastructure maintains full control of the data plane and offers to the service provider the possibility to exploit the logical adjacencies among its equipment. Another solution is that the service provider is offered the capability of controlling the isolated logical portion of the network. This paradigm is usually referred to as Infrastructure as a Service (IaaS) and has recently attracted the attention of many operators due to various advantages listed in [3].

In both the aforementioned networking applications, the process of building the virtual topology (i.e., the set of lightpaths established in the network) starting from the knowledge of the underlying physical infrastructure is a complex and relevant problem since it takes many different design parameters into account, and its effectiveness influences the quality of the service that the provider can eventually offer to its customers. Such a problem is commonly named *virtual topology design* (*VTD*). In general, it is possible to design different virtual topologies on top of the same infrastructure; in this paper, as has been already done in [4], we will focus on only one such virtual topology.

Manuscript received July 9, 2014; revised October 1, 2014; accepted October 2, 2014; published November 6, 2014 (Doc. ID 216655).

N. Fernández, R. J. Durán Barroso (e-mail: rduran@tel.uva.es), I. de Miguel, J. C. Aguado, and R. M. Lorenzo are with Universidad de Valladolid, Paseo de Belén, 15, 47011 Valladolid, Spain.

D. Siracusa (e-mail: domenico.siracusa@create-net.org), A. Francescon, and E. Salvadori are with CREATE-NET, Via alla Cascata 56/D, 38123 Trento, Italy.

http://dx.doi.org/10.1364/JOCN.7.00A162

Modern communication networks handle continuous variations of traffic. Therefore, it is possible that the VTD that has been designed at a given moment of the network's life is no longer suitable (it does not satisfy the requirements and objectives that have been set before). In such a case it is possible to resort to virtual topology *reconfiguration*, that is, the reconfiguration of some of the active lightpaths (potentially all) according to both the current network state (i.e., how is the virtual topology composed and what are the available resources) and the current traffic demand. The alternative policy to reconfiguration consists in having the lightpaths permanently established and modifying them only in exceptional cases, such as failures. This latter option is currently the most widespread, but it implies that the virtual topology is designed by overdimensioning the resources in operation, i.e., establishing more lightpaths than necessary, in order to avoid losses in cases of traffic peaks or failures of equipment, which clearly leads to an increment in the network costs. On the other side, reconfiguration offers many benefits [5] in terms of cost and energy savings, as well as on reducing the set of used resources, depending on the traffic demands. However, it has a disadvantage that should be carefully taken into account: When facing a reconfiguration event in IP over WDM networks, the modification of the optical connectivity of the nodes leads to temporal instability of the IP routing tables, and this issue may translate into packet losses. Hence, it is important to consider not only the most efficient virtual topology design for the current traffic pattern but also the analysis of the traffic disruption that can appear during the reconfiguration process.

The operations described so far require complex computations and the ability to foresee the traffic evolution. The computationally intensive operations (together with the need of keeping track of network configuration and network status) clearly calls for a centralized computation entity maintaining a database with the completed and ongoing lightpath establishments, supported by a protocol disseminating information about network configuration. On the other hand, *cognitive techniques* [6] can be exploited to make the computation entity capable of forecasting future traffic by learning from past history. Cognition can also be used for other purposes, for instance, to design the virtual topology and to support the reconfiguration process when it comes to deciding whether the current virtual topology should be kept or be reconfigured.

This paper presents and evaluates in a simulation environment the performance of a centralized control and management architecture in which a cognitive entity resorts to network knowledge, traffic forecasting, and memory to design and reconfigure virtual topologies. Moreover, the advantages of the cognitive methods proposed in this paper are also demonstrated in a more realistic environment by means of an emulated testbed, which includes the real implementation of generalized multiprotocol label switching (GMPLS) protocols for lightpath establishment. With respect to the work presented in [7], this paper proposes a new transition process from the old virtual topology to the new one, which helps decrease the losses incurred during the reconfiguration process and thus improves the results shown in [7], and an increased set of experimental results (both simulated and emulated) is shown.

The paper is organized as follows: in Section II the virtual topology reconfiguration process is described. The performance of the proposed methods is evaluated by means of simulation in Section III. Then, in Section IV, the methods are validated in the emulated testbed. Finally, Section V shows the main conclusions of the paper.

II. Cognitive Virtual Topology Design and Reconfiguration

VTD is an NP-complete problem that consists of three subproblems [8,9]:

- determining the set of lightpaths to be established in the network;
- assigning network resources for each lightpath, i.e., solving the routing and wavelength assignment (RWA) problem, or the routing and spectrum allocation problem; and
- routing the traffic through that set of lightpaths.

This problem can be solved by targeting different optimization objectives such as congestion (i.e., the traffic carried by the most loaded lightpath) [10–12], end-to-end delay [9,12,13], power consumption [14], or cost. Although there are trade-offs among those objectives, a network operator may be interested in the joint optimization of some of those parameters, that is, in solving a multiobjective problem. In those cases, the most interesting approaches aim at obtaining the Pareto optimal set (POS), i.e., a set of solutions where each solution (a virtual topology in our case) is characterized by the fact that it cannot be improved in terms of one of the optimization objectives without worsening the others.

One of the most important objectives when designing or optimizing an optical network is the cost. Network costs are usually divided into capital expenditures (CAPEX) and operational expenditures (OPEX). The latter are related to network usage, and thus include space rentals, energy consumption, network maintenance, and fault repair costs. While CAPEX should be taken into account in the provisioning stage of the network, OPEX can be minimized by the optimization of the network operation. Since communication networks handle continuous variations of traffic demands, some network elements can be temporarily powered off, thereby lowering OPEX due to reduced energy consumption.

In optical networks, the virtual topology can be statically configured, or the virtual topology can be reconfigured depending on both the network state and the current traffic demand to decrease cost and energy consumption, as well as to reduce the set of resources in operation. As previously mentioned, in reconfigurable optical networks, the design of the most efficient virtual topology for the current conditions is not the only aspect that should be taken into account, but the analysis of the traffic disruption that can appear during the reconfiguration process must also be considered. Hence, there are three important issues involved in the reconfiguration process (see [15] and references therein):

- determining a virtual topology that can efficiently transport the new traffic demand;
- deciding when to trigger a reconfiguration; and
- choosing the transition sequence, i.e., the time-ordered list of operations to be preformed to migrate from the current virtual topology to the new one.

The first subproblem is similar but not equal to the static VTD problem, as the current established virtual topology should be taken into account to avoid/minimize the disruption caused by the reconfiguration.

In the reconfiguration process, so as to decide when to trigger a reconfiguration, the literature proposes approaches that can be classified into off-line [16] and on-line policies [4,17]. The off-line reconfiguration is based on predesigned virtual topologies according to an estimation of traffic evolution, and it provides the sequence of virtual topologies to be established in each moment [16]. In contrast, on-line policies analyze network traffic in real time to decide when to trigger the reconfiguration and to design virtual topologies adapted to that traffic [4,17].

Regarding the transition from the current virtual topology to the new one, there are two approaches. The first approach consists of making a soft transition, i.e., establishing or releasing lightpaths step by step to minimize the impact of each reconfiguration step. However, with this approach, the instability of the routing tables has a longer duration. In contrast, making an abrupt variation reduces the time in which the routing tables are nonupdated, but the disruption caused during the reconfiguration can be higher as less valid routes appear in the routing tables until they become stable [18].

The use of cognition in optical networks has been recently proposed [6,19,20] as a way to improve network performance by exploiting learning. In particular, VTD and reconfiguration processes can take advantage of these techniques, as we demonstrated in the FP7 EU CHRON project [21]. A Cognitive Heterogeneous Reconfigurable Optical Network (CHRON) [6,21] is based on a centralized architecture where a central entity, called the cognitive decision system (CDS), decides how to control network resources according to traffic demand and user requirements. For this objective, it relies on cognitive methods, which exploit their capability to learn from previous history, in order to optimize network performance. Then, a control and management system (CMS) configures the network devices according to the CDS decisions, provides updates about the network configuration and resource availability, and notifies of any malfunctioning or anomaly. The architecture also includes a network monitoring system (NMonS), which mainly consists of different monitors distributed in the network that provide both traffic status and optical performance measurements to the CDS.

The CHRON can work with either a static or a reconfigurable approach for the virtual topology, and also supports the dynamic establishment of private lightpaths (i.e., lightpaths not being part of the virtual topology). In fact, the joint support of a virtual topology and dynamic lightpaths, allowing for resource sharing among them, is one of the advantages of such a network. When the CHRON is configured to work with a reconfigurable virtual topology, it can reduce the OPEX while maintaining the packet loss ratio (PLR) under user-defined values. For this issue, the CHRON uses the reconfiguration method proposed in this paper, i.e., an on-line reconfiguration policy combined with a multiobjective genetic algorithm, called CONGA-VTD [22], to design the virtual topology. Additionally, two cognitive techniques have been included in the reconfiguration process to enhance its performance. First of all, by forecasting the future traffic demands, and second, by complementing the virtual topology design algorithm with a knowledge base (KB) where solutions successfully used in the past are stored for potential reuse in the future. Then, CONGA-VTD is able to use this information by employing a cognitive technique to select the most useful information in the KB for the current and future traffic conditions.

In the following subsections, we focus on these capabilities of the CHRON architecture, and analyze how different modules of the CDS work together to implement a virtual topology reconfiguration process. Finally, we will extend the work in [7,22] by proposing a new policy to determine the transition sequence from the current to the new virtual topology, which reduces network disruption, and thus, PLR.

A. VTD and Reconfiguration

The CHRON CDS uses two of its submodules [6] to support the establishment and reconfiguration of the virtual topology: the network planner and decision system (NPDM) module, and the VTD module.

The virtual topology reconfiguration process is periodically triggered by the NPDM module (we call that period a time-slot). Let us suppose that the current time slot is h. Then, this process should decide how to act in the next time slot, $h+1$. The process consists of three subtasks (as shown in Fig. 1):

- the traffic forecast subtask, which predicts the traffic in the near future;
- the VTD subtask, which determines a set of virtual topologies adapted to the expected traffic in the near future

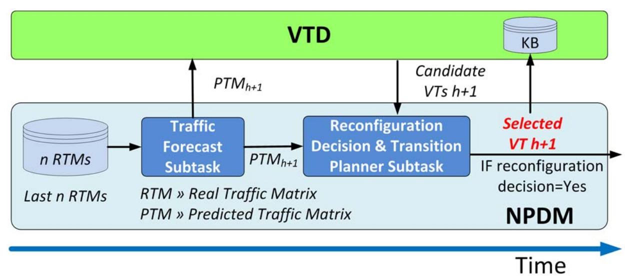

Fig. 1. Interaction between the NPDM and VTD modules of the CHRON [22]. Description of the virtual topology reconfiguration process.

(each with different trade-offs in terms of the optimization objectives); and

- the reconfiguration decision & transition planner subtask, which analyzes the candidate virtual topologies provided by the previous subtask, and determines whether the current virtual topology should be reconfigured or not, and how to do it.

The NPDM module relies on the real traffic matrixes (RTM) database, which is stored in the CDS. This database contains the set of n RTMs measured by means of the network monitors in the last n time-slots, i.e., from time-slot h to slot $h - (n - 1)$. These RTMs are inputs of the traffic forecast subtask. With that information, this subtask estimates the traffic matrix for the next time-slot, i.e., the predicted traffic matrix (PTM_{h+1}). Then, the VTD module designs a set of virtual topologies adapted to that predicted traffic, PTM_{h+1}. The virtual topologies provided by the VTD module are then evaluated by the reconfiguration decision & transition planner subtask, which makes the decision about keeping the same virtual topology or reconfiguring it considering the expected traffic PTM_{h+1}. Moreover, this subtask is in charge of selecting the virtual topology that will be established (in case of reconfiguration) among the ones proposed by the VTD module. Moreover, in case of reconfiguration, this subtask also determines the transition sequence from the current virtual topology to the selected one (selected VT_{h+1} in Fig. 1) to minimize the PLR. Then, the virtual topology is established in the physical network at slot $h + 1$ according to that sequence. When the system receives information from the traffic monitors, the real traffic (RTM) in slot $h + 1$ is built and the real traffic losses and the real OPEX are evaluated. Moreover, the selected virtual topology is introduced in a knowledge base (KB), which can be used by the VTD module to design new solutions in the future (as will be explained later).

In the following subsections we describe the different subtasks in more detail.

B. Traffic Forecast Subtask

The traffic forecast subtask is performed by the NPDM. This subtask queries the RTM database (i.e., the set of the last n RTMs) as inputs, and it returns a forecast of the traffic matrices for the next time slots. In this work we have considered just the forecast related to the following slot ($h + 1$). In particular, the autoregressive integrated moving averages (ARIMA) technique has been implemented for this aim. ARIMA is a statistical model that uses variations and data regressions to find patterns to model data or to forecast future data [23]. By means of this technique, the NPDM is able to forecast the traffic in the near future and, thus, is able to make the best decision about the reconfiguration of the virtual topology.

C. VTD Subtask

The VTD module uses a genetic algorithm, called CONGA-VTD [22], which is able to solve the three sub-problems of the VTD problem. CONGA-VTD is a multi-objective (three-objective) algorithm to design virtual topologies with the aim of maximizing network capacity (by minimizing the congestion), minimizing the OPEX, and reducing the number of different lightpaths between the current virtual topology and the one proposed for the next slot (to minimize the reconfiguration disruption).

Like other (but not all) multiobjective algorithms, CONGA-VTD is targeted to obtain an estimate of the POS. In the set of solutions that CONGA-VTD returns, the virtual topology that is currently established is also included (as it is the one with no changes between the current and the new virtual topology, and thus the optimal solution in terms of one of the three objectives). Hence, keeping that topology (i.e., not reconfiguring) is also a candidate choice for the next time slot.

To estimate OPEX, we use the techno-economic model presented in [24], which takes into account the costs of equipment such as transponders, erbium-doped fiber amplifiers, optical cross-connects (OXCs), and IP/MPLS nodes, in terms of power consumption costs, rental costs, and maintenance and repair costs. It should be noted that the space rental cannot be optimized during network operation, as it depends on all the equipment installed in the network (idle and in operation).

A genetic algorithm is, in principle, incapable of acquiring and using knowledge from past executions. However, works in cognitive radio like [25] have demonstrated that learning can be included in genetic algorithms in order to select the initial population of the genetic algorithm, leading to improvements on its performance. Following that idea, CONGA-VTD exploits past knowledge by employing some of the virtual topologies used in the past as starting points of the genetic algorithm. To acquire that knowledge, when the network is reconfigured with a virtual topology proposed by CONGA-VTD, that virtual topology is stored in a KB. Then, when the method is launched again to find new virtual topologies for a future time slot, a reasoning process is applied. All the virtual topologies in the KB are evaluated according to the current network state and estimated traffic matrix, and those solutions composing the POS according to the three objectives of CONGA-VTD (i.e., those virtual topologies that better fit to the current traffic and network state) are included in the initial population of the genetic algorithm. The initial population also includes a set of randomly generated virtual topologies and other special ones, calculated *ad hoc*, as described in [22]. Thanks to the use of this cognitive technique, CONGA-VTD finds better solutions in less time.

On the other hand, additional cognition is incorporated in the algorithm. CONGA-VTD designs virtual topologies in which all the lightpaths fulfill the quality of transmission (QoT) requirements posed by the user. To estimate the QoT of the lightpaths of the candidate virtual topologies, the cognitive QoT estimator proposed in [26] is used. This QoT estimator employs a case-based reasoning (CBR) technique, and it is included at the end of genetic loop of CONGA-VTD. Moreover, during the execution of the genetic loop, CONGA-VTD uses a tabu list to avoid the use of those lightpath configurations that do not fulfill the QoT requirements when solving the routing and wavelength assignment problem. This tabu list is dynamically updated with the solutions provided by the CBR method for the lightpath configurations of previously analyzed traffic matrices. This combination of the use of the CBR technique with the tabu list accelerates the execution of CONGA-VTD without degrading performance (see [27] for details).

D. Reconfiguration Decision & Transition Planner Subtask

The reconfiguration decision & transition planner subtask determines whether the reconfiguration should be applied or not (by considering the candidate virtual topologies returned by the VTD module), and in case of reconfiguration, how to do it. In particular, when assessing a virtual topology for a potential reconfiguration, it takes into account the traffic that would be temporarily disrupted because of the temporal instability of the routing tables.

It should be noted that, when the VTD problem is solved, the VTD module also estimates how the predicted traffic will be routed on the new virtual topology. However, the real update of the routing tables of the nodes will be done by a routing protocol like OSPF, and for that reason, due to the timings involved in that protocol, there will be a temporal instability of the routing tables potentially leading to the PLR.

To estimate the PLR, we have adopted a very conservative approach that assumes that all the traffic traversing disrupted lightpaths is lost until the routing tables become stable. On the other hand, we have made the assumption that when two or more lightpaths are established from node s to node d and any of these lightpaths is removed, all the traffic that previously used the removed lightpaths is sent through the rest of the lightpaths between nodes s and node d as long as there is enough capacity in them. Finally, if a lightpath has to transport more traffic than its capacity, the exceeding traffic is considered lost traffic.

The PLR occurs mainly during the reconfiguration process, specifically when a lightpath of the current virtual topology has to be released (since it does not appear in the new one) and there is not any other optical circuit between the considered pair of nodes. In this case, the traffic between those nodes will be carried following a multihop route in the new virtual topology (i.e., traversing more than one lightpath) once the routing tables are stable. However, in the meantime, there will be traffic losses while the routing tables are being updated.

To minimize this problem, we propose a transition process that takes into account the behavior of the upper layers with the aim of reducing packet losses. The key idea does not consist in performing an abrupt reconfiguration, but in executing the reconfiguration process as a set of consecutive phases. First of all, those new lightpaths that will be present in the new virtual topology are established (if possible), followed by the establishment of a set of temporal lightpaths that will help in the transition process (but will be later removed). Then, the reconfiguration process waits until the routing tables are updated according to that intermediate new situation, and, finally, those lightpaths not being part of the new virtual topology are removed.

The procedure to select which temporal lightpaths to establish is as follows. Each entry of the routing table at a node associates a set of destination IP addresses to an output port, and each output port is connected to a lightpath. Let us assume a lightpath connecting nodes a and b ($a \to b$) is going to be removed when reconfiguring from the current to the new virtual topology (we will call such a lightpath “affected lightpath”). Thus, in the final virtual topology, traffic routed from a to b will be routed through several lightpaths and thus will traverse several intermediate nodes, e.g., $a \to n_1 \to n_2 \to \cdots \to n_n \to b$. In this case, a temporal lightpath would be established (if possible) between the source node of the “affected lightpath” (node a) and the first intermediate node that will handle that traffic in the new virtual topology (node n_1). Note that, in fact, there will be at least one lightpath between those two nodes (a and n_1) in the new virtual topology, so one may wonder why to add this temporal lightpath. The reason to add that temporal lightpath is that the output port of node a, which was connected to the affected lightpath, will now be connected to this temporal lightpath, so that the traffic being sent to b will immediately flow through this new lightpath as soon as it is established. In this way, the affected lightpath can be released without causing an increment in the PLR, as it will carry no traffic when removed (except if some lightpath becomes temporarily congested during this process, although this traffic loss will be definitely smaller than the one experienced without recurring to this process). Finally, when the routing tables become stable, this temporal lightpath can be removed as there is, at least, another lightpath connecting these two nodes in the virtual topology designed by the VTD module.

In this way, the transition process consists of seven steps:

1. Establish, if possible, all the new lightpaths required in the new virtual topology (using the route and the wavelength determined by the VTD module).
2. For those lightpaths that will appear in the final virtual topology (the one designed by the VTD module) but could not be established in step 1 (as they require resources currently in use by the existing virtual topology), try to establish them by using a different wavelength than the one planned by the VTD module. However, these lightpaths cannot use resources assigned by the VTD module to other lightpaths in the new virtual topology. (Note that these lightpaths are not the definitive ones since they do not have the final wavelength designed by the VTD module, and therefore they will be later re-established on that wavelength in steps 6 and 7.)
3. Establish (if possible) temporal lightpaths for the affected lightpaths as previously explained (before the

description of this transition process). Again, these temporal lightpaths cannot use any of the resources assigned by the VTD module to other lightpaths in the new virtual topology. Note that in step 2 and step 3 temporal lightpaths are established to minimize traffic losses during the reconfiguration process. However, in step 2, those lightpaths will be part of the final virtual topology (but in a different wavelength), while in step 3 these lightpaths will not be part at all of the final virtual topology, but they are useful to reduce losses in the transition process.

4. Wait until the routing tables become stable.
5. Release all the lightpaths from the current virtual topology that are not included in the new virtual topology (i.e., the only lightpaths that will remain in the virtual topology of the network are those established in steps 1, 2, and 3). Traffic losses appearing in the reconfiguration process are due to this step.
6. Establish the remaining lightpaths of the new virtual topology that have not been established in step 1 following the route and the wavelength determined by the VTD module.
7. Remove the lightpaths established in steps 2 and 3. This is the final step of the transition process.

Therefore, the complete operation of the reconfiguration decision & transition planner subtask is as follows. It evaluates all the virtual topologies proposed by the VTD module (including the currently established one) in terms of PLR and OPEX, given a forecasted traffic demand related to the next time slot. To estimate PLR, the module takes into account the transition process that we have just described. Then, this module chooses a virtual topology among all the candidate possibilities. The selection of the most suitable virtual topology is done according to a nested metric: first, the topologies generating the lowest loss (PLR) are chosen; in case of ties, the topology providing the lowest OPEX is selected. If the selected virtual topology is different from the one currently established, the reconfiguration is executed in the network (using the transition process described above).

III. Performance Evaluation by Means of Simulations

The performance of the new virtual topology reconfiguration process has been analyzed by means of simulation and then validated by means of emulation. In these tests, the 14-node Deutsche Telecom (DT) network has been used as the physical network, assuming that each cable consists of two fibers, one per direction. A 32 fixed-grid wavelength channel configuration in each fiber has been considered. Moreover, the physical layer impairments have been taken into account to ensure that all lightpaths of the virtual topologies comply with QoT requirements.

In the simulator, each node is equipped with 192 10G transponders (each one composed of a transmitter and a receiver). The OPEX are evaluated using the techno-economic model presented in [24]. The reference unit for the cost is the cost of one 10G transponder. The traffic has been generated following the model proposed by Gençata and Mukherjee [4] (which has also been used in other studies about reconfiguration). It considers that the traffic between each pair of nodes (from node s to node d) at time t (in seconds), $\lambda^{s,d}(t)$, is given by

$$\lambda^{s,d}(t) = \Lambda^{s,d} \cdot \beta(t)\left[1 + \frac{1}{2}\,\sin\left(\frac{2\cdot\pi\cdot t}{86400}\right)\right], \tag{1}$$

where $\Lambda^{s,d}$ is the average traffic from s to d in one day, i.e., 86,400 s. $\beta(t)$ adds the bursty character to the traffic demand and it is randomly generated each time that this function is invoked using a uniform random variable between $[1-\varepsilon, 1+\varepsilon]$, where ε is a user-defined parameter that controls the amount of burstiness. In this paper, $\Lambda^{s,d}$ has been generated randomly using a uniform distribution with different average values for all source–destination pairs, and the degree of burstiness β has been set to 5%. The time-slot duration (h) has been set to 1 h. The virtual topology reconfiguration process has been evaluated considering the evolution of the network along four (simulated) days. The results obtained in the paper for different parameters have been averaged over 50 simulations with different values of $\Lambda^{s,d}$. Results are shown with 95% confidence intervals.

The results of the proposed reconfiguration policy have been compared with those provided by a static virtual topology and a reconfigurable virtual topology employing the reconfiguration method proposed by Gençata and Mukherjee [4]. In this method, only one lightpath is established or released in each time slot depending on the traffic carried by the most loaded lightpath. To obtain a fair comparison, at the beginning of each simulation, the three methods (static, Gençata, and the proposed method—labeled CHRON) configure the network with the same virtual topology that was designed by CONGA-VTD assuming that the traffic between each pair of nodes was $\Lambda^{s,d}\cdot(1+0.05)\cdot(1+0.5)$, i.e., the maximum possible traffic according to the model of Eq. (1). As CONGA-VTD obtains a set of virtual topologies, the choice among them for this initial virtual topology is that with the lowest OPEX.

To estimate the predicted traffic matrix (PTM_{h+1}), the traffic forecast process has been configured to use the RTMs from the last 48 time slots. Moreover, it has been considered that OSPF requires 60 s to update the IP routing tables. (This value has been adopted after a number of simulation experiments for an IP-over-WDM network using OSPF as the routing protocol with the OPNET Modeler simulation platform.)

First of all, we have analyzed the advantages of the transition method proposed in Subsection II.D. For that aim, we have considered the Gençata and Mukherjee reconfiguration policy and have executed it directly, also using the transition mechanism. The reason to use initially that algorithm (instead of the CHRON cognitive techniques) is to emphasize that the technique can be used in combination with other VTD and reconfiguration methods, and to show its advantages without the inclusion of any cognitive technique. The average and the maximum PLRs obtained when

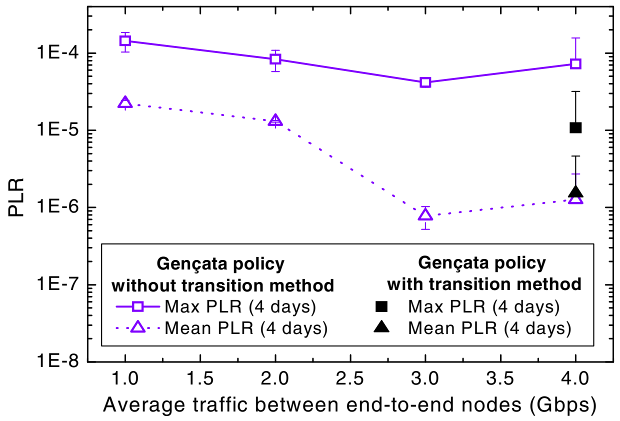

Fig. 2. PLR of the CHRON using the Gençata and Mukherjee reconfiguration policy [4] with and without the new transition method.

activating/deactivating the transition method are shown in Fig. 2. As shown in that figure, no PLR was observed when the transition method was employed (except for very high traffic load). In contrast, if the transition method is not used, the PLR is observed. Regarding this figure, one may wonder why the PLR does not increase with the average traffic load. It should be noted that the initial virtual topology is designed independently for each traffic load. Therefore, for higher traffic loads, the virtual topology employs a higher number of resources to avoid a PLR increase; that is, the percentage of the traffic carried by the lightpaths compared to their capacity is similar, independently of the traffic load. On the other hand, the Gençata and Mukherjee method makes a single change during each reconfiguration and, therefore, the PLR produced by these reconfigurations is almost constant.

Moreover, to show the advantages of the transition process also in the CHRON environment, Fig. 3 shows the comparison in terms of OPEX when the new transition method is used or not, for different average traffic loads. OPEX is shown instead of the PLR as the CHRON reconfiguration method has been developed to reduce the network OPEX while maintaining the PLR under acceptable values. As can be seen in Fig. 3, the use of the transition method leads to a better adaptation of the virtual topology to the traffic evolution and a reduction in the OPEX for all traffic loads. Obviously, the difference is more noticeable when the traffic load is low (up to 10%) as the network requires a small amount of its resources to carry the traffic demand and can use configurations that reduce OPEX while maintaining the PLR under acceptable values.

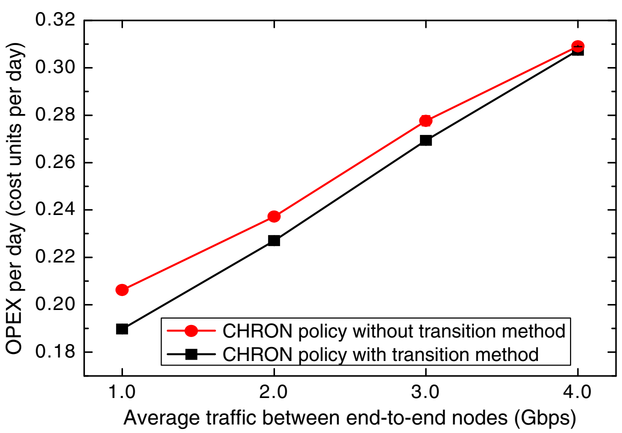

Fig. 3. OPEX of the CHRON using the CHRON reconfiguration policy with and without the new transition method.

Once the advantages of the transition method have been demonstrated, the performance of the whole reconfiguration method (i.e., including traffic forecasting, the CONGA-VTD method, and the reconfiguration decision & transition planner mechanism) has been evaluated. To make a fair comparison, both our proposal and the Gençata and Mukherjee policy have used the transition method proposed in Subsection II.D.

The evolution of OPEX and PLR with time is shown in Fig. 4 for an average traffic load between each pair of nodes of 1 Gbps, and in Fig. 5 for a traffic load of 3 Gbps. The PLR shown in these figures represents the average PLR during 1 h, i.e., during a time slot. Results show that the CHRON reconfiguration method achieves lower OPEX than when a static virtual topology is used or when the reconfigurable virtual topologies provided by the Gençata and Mukherjee

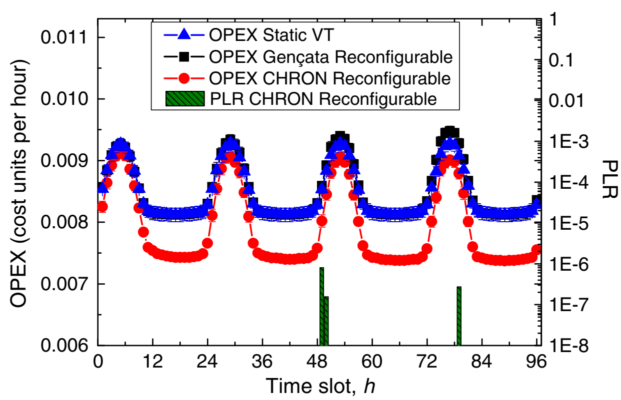

Fig. 4. OPEX and PLR per time slot for an average traffic of 1 Gbps between each pair of nodes.

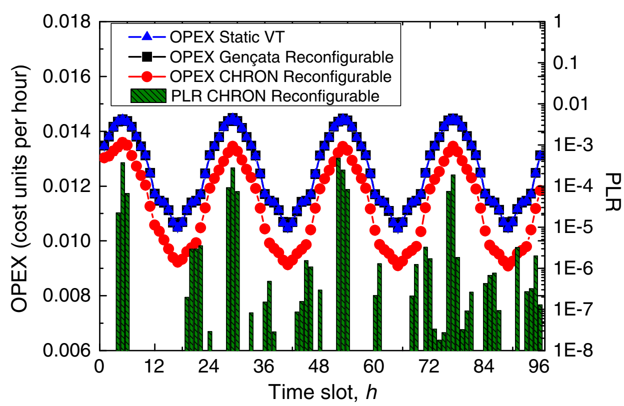

Fig. 5. OPEX and PLR per time slot for an average traffic of 3 Gbps between each pair of nodes.

method are employed. The savings provided by the CHRON reconfiguration method are located mainly in the hours with less traffic demand. However, due to the reconfiguration, some losses appear when using the CHRON technique, although in both cases (Figs. 4 and 5) the PLR is below 10^{-3}. In contrast, the Gençata and Mukherjee policy has no PLR in this scenario due to the minor variation in the virtual topologies that it introduces and also due to the use of the transition method described in Subsection II.D (as shown in Fig. 2). It is worth mentioning that, for the Gençata and Mukherjee method, we also performed additional simulations employing smaller time slots (thus enabling a higher number of changes of the virtual topology in 1 h), but worse performance was obtained, as the PLR increases without bringing advantages in terms of OPEX.

We have also analyzed the impact of different average traffic loads, between nodes (from 1 to 4 Gbps) in terms of PLR (Fig. 6), OPEX (Fig. 7), and occupation factor, i.e., the percentage of wavelength channels used (Fig. 8). Three PLR values are shown in Fig. 6: the average PLR during the four days (this value is calculated according to the total losses and the total traffic injected in the network during the analyzed time), the maximum value of the PLR calculated in the four days (for a certain hour, i.e., time slot), and the maximum value of the PLR calculated in the last day (for a certain hour).

Results show that the static virtual topology and the reconfigurable network that adopts the Gençata and Mukherjee method do not produce traffic losses for the average traffic between nodes of 1, 2, and 3 Gbps. However, when the traffic increases to 4 Gbps, some traffic losses appear because the virtual topology is not able to carry all the required traffic demands.

Regarding the CHRON reconfiguration method, the maximum PLR that appears in an event of reconfiguration is lower than 10^{-3} when the traffic between each pair of nodes is lower than 3 Gbps. Obviously, the average PLR

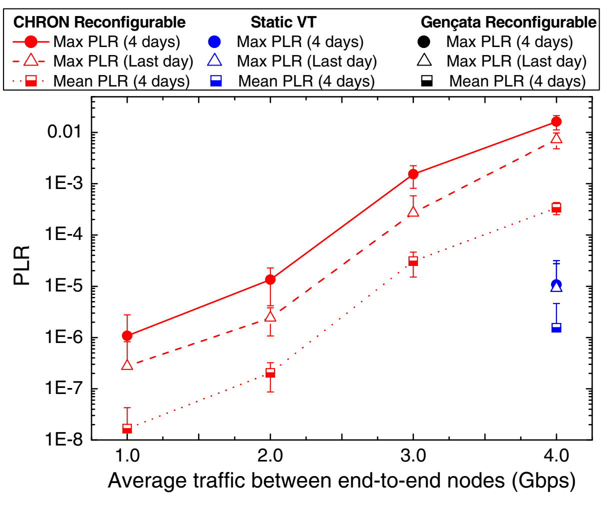

Fig. 6. PLR along four simulated days for different average traffic loads with the static virtual topology, the Gençata and Mukherjee reconfigurable network, and the CHRON reconfiguration method.

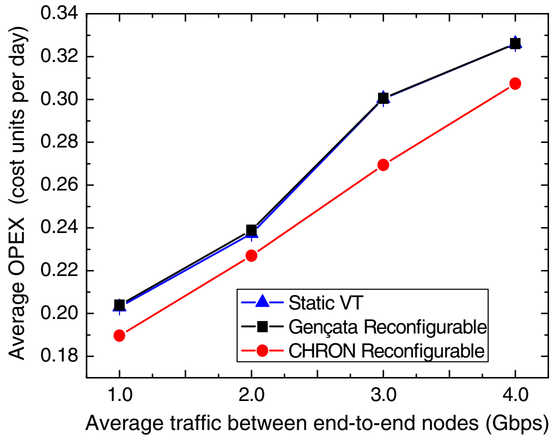

Fig. 7. Average OPEX per day for different average traffic loads with the static virtual topology, the Gençata and Mukherjee reconfigurable network, and the CHRON reconfiguration method.

is much lower than this value. Moreover, the maximum value of the PLR found in the last day of simulations is lower than that value found when the maximum values found during the four days are averaged. This demonstrates that the VTD algorithm used in the CHRON reconfiguration method learns from past experience and it can reduce the PLR generated during the reconfiguration process by using the knowledge acquired in the previous days.

The main advantages of the CHRON reconfiguration method come in terms of reducing OPEX (Fig. 7) and occupation factor (Fig. 8). As shown in Fig. 7, the CHRON reconfiguration method is the mechanism that achieves the lowest values of average OPEX accumulated along one day. This reduction is around 4.35% in the less favorable case (2 Gbps) and more than 10% in the best scenario (3 Gbps).

Then, as can be seen in Fig. 8, the occupation factor in the CHRON reconfigurable virtual topology leads to a reduction of 5.47 percentage points (in the worst case) in occupation if compared to a static solution. Furthermore,

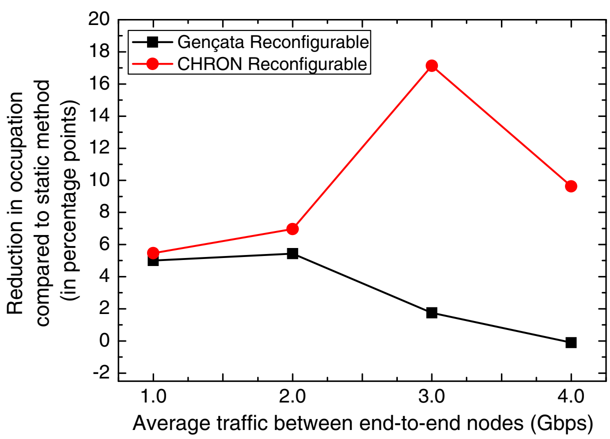

Fig. 8. Reduction in occupation for different average traffic loads when the Gençata and Mukherjee reconfigurable network and the CHRON reconfiguration method are compared with the use of a static virtual topology.

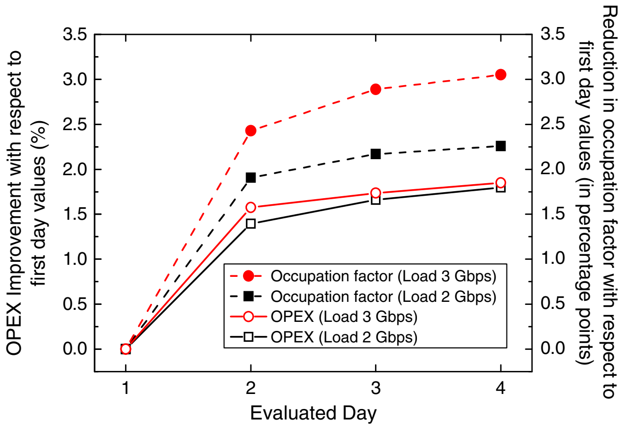

Fig. 9. Improvement evolution in terms of OPEX and occupation factor during the emulation for two different traffic loads.

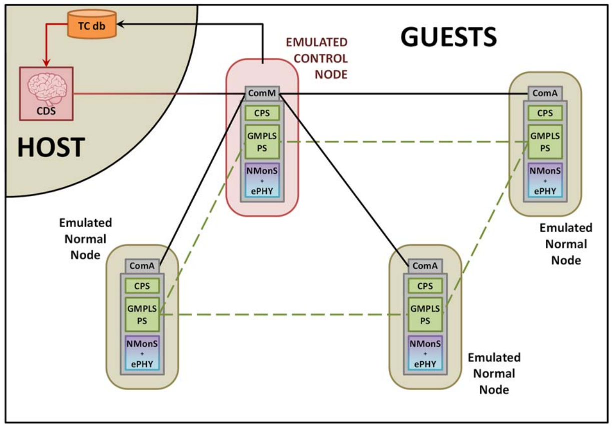

Fig. 10. CHRON emulated testbed architecture in a sample network composed of four nodes.

it can reach an improvement of 17 percentage points compared with the static network for an average traffic load of 3 Gbps between each pair of nodes. The reduction in occupation is more significant than that obtained by the Gençata and Mukherjee method, as shown in Fig. 8. The advantages of reducing the occupation factor is that those resources not used by the virtual topology can be used for other services offered by the operator (e.g., dynamic lightpaths on demand or another virtual topology), thus increasing the overall performance of the network.

To show the advantages of using the cognitive techniques implemented in the CHRON reconfiguration process, both the OPEX improvements achieved every day (in average) with respect to the values of the first day and the reduction in the occupation factor for two different traffic loads are shown in Fig. 9. It is important to note that, at the beginning of the simulations, the KBs are empty except the one that feeds the traffic forecast subtask, which keeps the traffic values during the last 48 h, (i.e., enough to make a good traffic prediction). Therefore, the advantages shown are due to the cognitive techniques implemented excluding the traffic forecast. Moreover, the values of the first day are also benefited by cognition as they are the average of the results obtained during the complete day (i.e., the first 24 time slots). Results in Fig. 9 show that the CHRON learns from the past decisions that it made, and it is able to design virtual topologies and apply the reconfiguration policy in a more effective way as the knowledge acquired grows. This behavior is noticeable in terms of both objectives and for all the traffic loads.

In summary, this simulation study has pointed out that the CHRON virtual topology reconfiguration outperforms the other proposals as it reduces the OPEX and the occupation factor while maintaining the PLR under an assumable value.

IV. Performance Evaluation by Means of Emulation

The behavior of the CHRON reconfiguration process has also been demonstrated in a “near real” environment: the CHRON emulated testbed.

As shown in Fig. 10, the CHRON emulated testbed architecture is composed of a decision-making control node and many normal nodes (three nodes in the figure). Each normal node is composed of the following four main building blocks:

- Communication agents (ComA): the interface between the modules of the normal nodes and the CDS located in the control node.
- Control plane scripts (CPS): a set of scripts to ease the interfacing of the GMPLS modules with the rest of the CHRON system.
- GMPLS protocol system (GMPLS PS): mainly composed of the OSPF-TE and RSVP-TE protocols, used to disseminate network information and perform lightpath provisioning, respectively.
- Network monitoring system (NMonS) agents (e.g., traffic monitor) and physical layer modules (ePHY) including the physical layer manager and the emulated devices.

In addition to these modules, the control node includes the following elements:

- Communication master (ComM): the counterpart of the ComA.
- Topology and configuration manager (TCMG): a server for the inputs provided by the GMPLS protocols and the NMonS agents.
- Topology and configuration database (TC db): it stores the information related to the network topology, the configuration of the devices, and the monitored parameters.
- Cognitive decision system (CDS): the brain of the cognitive optical network, already described.

All the aforementioned modules have been designed and implemented for the CHRON testbed, except for the GMPLS PS module, whose core software is actually based on the Dragon GMPLS framework [28], an open-source C/C++ software relying on GNU Zebra routing software [29]. The Dragon framework provides a suite of software modules implementing the main components of a GMPLS control plane, i.e., OSPF-TE and RSVP-TE protocols. Such protocols have been adequately extended to carry some additional information about device configuration and network status.

The CHRON emulator is running on a single server. The emulated control node and the normal nodes are running in guest virtual machines, while the CDS modules and the TC db are directly installed in the host machine (to provide it an adequate amount of computational capacity and to mimic a situation in which the CDS is installed in a dedicated machine), as shown in Fig. 10. The emulation server is a Dell PowerEdge R310 with 32 GB RAM equipped with a 4-core Intel Xeon X3480, 3.06 GHz and 2 TB hard disk.

The traffic has been generated using the Gençata and Mukherjee model shown in Eq. (1). The results have been collected in four days of emulations and, as in the simulations, the time-slot duration has been set to 1 h. Each point in the following figures is the average of the results obtained using five different values of $\Lambda^{s,d}$ randomly generated following a uniform traffic distribution between each pair of nodes with mean 1 Gbps.

Figure 11 shows the number of lightpaths of the virtual topology established in each time slot. Moreover, it also shows the time required to complete the reconfiguration process. Note that, according to Eq. (1), the maximum traffic is injected in the network at hours 6, 30, 54, and 78, while the minimum values correspond to hours 18, 42, 66, and 90. As Fig. 11 shows, the number of lightpaths (and so the resources in operation) gradually increases and decreases adapting to traffic variations. Regarding the reconfiguration time, it consists of three different components.

- VTD calculation time: the time that the VTD module requires to obtain the set of virtual topologies, that is, the estimation of the POS. As shown in Fig. 11, this time in the emulations is around 800 s. However, it should be lower in a real CDS, as the entire network was emulated in the same machine and the multithreading capacity of the CDS has been disabled during the emulations (for instance, when simulations were performed and multithreading was used, the computing time was around 130 s on a Debian GNU/Linux machine equipped with four AMD Opteron 6172 2.1 GHz processors and 64 GB RAM memory). Moreover, it is important to remark that this value includes the time required to forecast the traffic and the time elapsed in the evaluation of the QoT of all the lightpaths of all virtual topologies designed by the VTD module.
- "Make reconfiguration" decision time: the time required to evaluate the PLR of the virtual topologies provided by the VTD module (and also to determine the transition sequence from the current virtual topology to the new one), to decide whether to reconfigure or not, and to choose the virtual topology to be implemented. Figure 11 shows that this time is less than 60 s. This time can be also reduced using multithreading capacity.
- CMS time: the time the CMS needs to set up and tear down (by means of the RSVP-TE protocol) lightpaths to evolve from the current to the new virtual topology, following the transition sequence described in Subsection II.D. The analysis of this time component is the most important objective of the emulation study. The emulated testbed results show that this time is less than 240 s (including the time required for the IP routing tables to update in steps 4 and after step 7 of the transition method).

In summary of the timing analysis shown in Fig. 11, the CHRON reconfiguration strategy can be used in a real network (considering a time slot duration of 1 h) as the complete reconfiguration time is less than 20 min. The CMS time depends on the number of changes (lightpaths added or released) from the existing virtual topology to the new one, but an important component is also the time required to update the IP routing tables.

Figure 12 shows the evolution of the OPEX and the average PLR per time slot (i.e., 1 h) during the emulation time. In these emulation tests, no traffic losses were observed even when the virtual topology was periodically reconfigured (when needed) to reduce the OPEX. Such good performance is due to the transition sequence proposed in

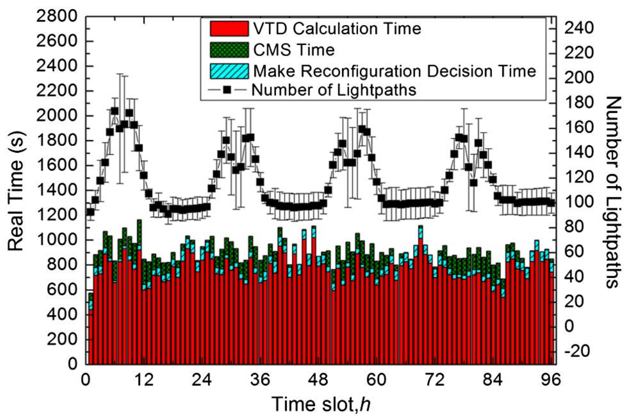

Fig. 11. Evolution of the number of lightpaths established in the CHRON emulated network of the virtual topologies and the time required to reconfigure the network in each time slot when the mean node-to-node traffic load is 1 Gbps.

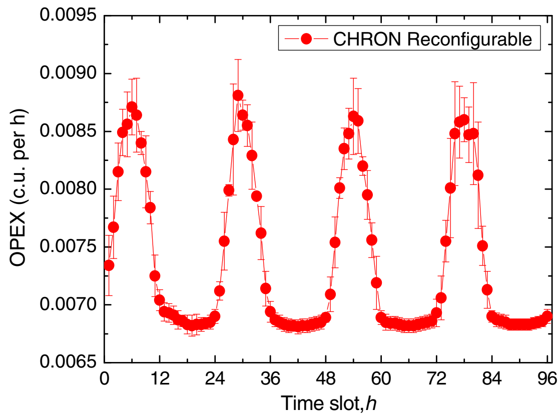

Fig. 12. OPEX per hour of the virtual topologies established in the CHRON emulated network and average PLR in each time slot when the mean node-to-node traffic load is 1 Gbps (no losses appear for this traffic load).

Subsection II.D. Regarding the network OPEX, the behavior and the values obtained in the emulation study are similar to those obtained with the simulations.

In summary, the CHRON reconfiguration method successfully enables the periodic adaptation of the virtual topology embedded in the network, reducing the OPEX (mainly in hours with low traffic) without causing significant PLR.

V. Conclusion

A new virtual topology reconfiguration method has been presented in this paper. This mechanism utilizes cognitive techniques to improve the overall performance of the network. These techniques include traffic forecasting, a multiobjective virtual topology design algorithm, and a reconfiguration policy. Moreover, a new reconfiguration transition sequence is introduced in this paper to reduce the traffic losses due to the instability of the routing tables during the reconfiguration process.

The performance of the new proposal has been analyzed and compared with previous works by means of simulation, and, then, the operation of the new technique has been demonstrated in a near real environment by means of an emulated testbed.

As a conclusion, the reconfiguration of the virtual topology using the CHRON reconfiguration policy is a useful technique that can be implemented in a real network and reduces the OPEX and the occupation factor while maintaining the PLR under low values.

Acknowledgments

This research has been partially supported by the CHRON project, with funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 258644. Natalia Fernández would like to thank the Council of Education of the Regional Government of Castilla-León and the European Social Fund for their support.

References

[1] I. Tomkos, P. Zakynthinos, D. Klonidis, D. Marom, S. Sygletos, A. Ellis, E. Salvadori, D. Siracusa, M. Angelou, G. Papastergiou, N. Psaila, J. F. Ferran, S. Ben-Ezra, F. Jimenez, and J. P. Fernández-Palacios, "Spatial-spectral flexible optical networking: Enabling switching solutions for a simplified and efficient SDM network platform," *Proc. SPIE*, vol. 9009, 90090H, 2014.

[2] O. Gerstel, M. Jinno, A. Lord, and S. J. B. Yoo, "Elastic optical networking: A new dawn for the optical layer?" *IEEE Commun. Mag.*, vol. 50, no. 2, pp. s12–s20, 2012.

[3] J. Carapinha and J. Jiménez, "Network virtualization—A view from the bottom," in *Proc. 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures (VISA)*, Barcelona, Spain, 2009, pp. 73–80.

[4] A. Gençata and B. Mukherjee, "Virtual topology adaptation for WDM mesh networks under dynamic traffic," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 236–247, 2003.

[5] W. Golab and R. Boutaba, "Policy-driven automated reconfiguration for performance management in WDM optical networks," *IEEE Commun. Mag.*, vol. 42, no 1, pp. 44–51, 2004.

[6] I. de Miguel, R. J. Durán, T. Jiménez, N. Fernández, J. C. Aguado, R. M. Lorenzo, A. Caballero, I. Tafur, Y. Ye, A. Tymecki, I. Tomkos, M. Angelou, D. Klonidis, A. Francescon, D. Siracusa, and E. Salvadori, "Cognitive dynamic optical networks [Invited]," *J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A107–A118, Oct. 2013.

[7] D. Siracusa, A. Francescon, N. Fernández, I. de Miguel, R. J. Durán Barroso, J. C. Aguado, and E. Salvadori, "Experimental evaluation of virtual topology design and reconfiguration in optical networks by means of cognition," in *Proc. OFC*, 2014, paper W3H.5.

[8] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee, "Some principles for designing a wide-area WDM optical network," *IEE/ACM Trans. Netw.*, vol. 4, no. 5, pp. 684–696, Oct. 1996.

[9] S. Ghose, R. Kumar, N. Banerjee, and R. Datta, "Multihop virtual topology design in WDM optical networks for self-similar traffic," *Photon. Netw. Commun.*, vol. 10, no. 2, pp. 199–214, Sept. 2005.

[10] R. J. Durán, I. de Miguel, N. Merayo, P. Fernandez, F. Lago, J. Aguado, J. Blas, R. Lorenzo, and E. Abril, "Multipurpose genetic algorithm based in Pareto optimality to design logical topologies in reconfigurable WDM networks," in *Proc. Int. Conf. Transparent Optical Networks (ICTON)*, 2006, vol. 4, pp. 27–30.

[11] R. J. Durán, I. de Miguel, N. Merayo, P. Fernández, J. C. Aguado, R. M. Lorenzo, and E. J. Abril, "Multiobjective genetic algorithm to design cost-efficient wavelength-routed optical networks," in *Proc. Int. Conf. Transparent Optical Networks (ICTON)*, July 1–5, 2007, vol. 4, pp. 96–99.

[12] R. Ramaswami and K. Sivarajan, "Design of logical topologies for wavelength-routed optical networks," *IEEE J. Sel. Areas Commun.*, vol. 14, no. 5, pp. 840–851, 1996.

[13] R. J. Durán, I. de Miguel, N. Merayo, P. Fernández, R. M. Lorenzo, and E. J. Abril, "Minimisation of end-to-end delay in reconfigurable WDM networks using genetic algorithms," *Eur. Trans. Telecommun.*, vol. 20, pp. 722–733, Dec. 2009.

[14] N. Fernández, R. J. Durán, I. de Miguel, N. Merayo, D. Sánchez, M. Angelou, J. C. Aguado, P. Fernández, T. Jiménez, R. M. Lorenzo, I. Tomkos, and E. J. Abril, "Cognition to design energetically efficient and impairment aware virtual topologies for optical networks," in *Proc. 16th Int. Conf. on Optical Network Design and Modeling (ONDM)*, Apr. 17–20, 2012.

[15] J. Wu, "A survey of WDM network reconfiguration: Strategies and triggering methods," *Comput. Netw.*, vol. 55, no. 11, pp. 2622–2645, 2011.

[16] I. Baldine and G. Rouskas, "Traffic adaptive WDM networks: A study of reconfiguration issues," *J. Lightwave Technol.*, vol. 19, no. 4, pp. 433–455, Apr. 2001.

[17] R. J. Durán, I. de Miguel, N. Merayo, P. Fernández, J. Aguado, R. M. Lorenzo, and E. J. Abril, "A reconfiguration policy to minimize packet loss rate in traffic adaptive WDM," in *Proc. 12th European Conf. on Networks and Optical Communications (NOC)*, Stockholm, Sweden, 2007, pp. 618–625.

[18] G. N. Rouskas and M. H. Ammar, "Dynamic reconfiguration in multihop WDM networks," *J. High Speed Netw.*, vol. 4, no. 3, pp. 221–238, 1995.

[19] G. S. Zervas and D. Simeonidou, "Cognitive optical networks: Need, requirements and architecture," in *Proc. Int. Conf. on Transparent Optical Networks (ICTON)*, Munich, Germany, July 2010, paper We.C1.3.

[20] W. Wei, C. Wang, and J. Yu, “Cognitive optical networks: Key drivers, enabling techniques, and adaptive bandwidth services,” *IEEE Commun. Mag.*, vol. 50, no. 1, pp. 106–113, Jan. 2012.

[21] A. Caballero, R. Borkowski, I. de Miguel, R. J. Durán, J. C. Aguado, N. Fernández, T. Jiménez, I. Rodríguez, D. Sánchez, R. M. Lorenzo, D. Klonidis, E. Palkopoulou, N. P. Diamantopoulos, I. Tomkos, D. Siracusa, A. Francescon, E. Salvadori, Y. Ye, J. López, F. Pittalà, A. Tymecki, and I. Tafur, “Cognitive, Heterogeneous and Reconfigurable Optical Networks: The CHRON Project,” *J. Lightwave Technol.*, vol. 32, no. 13, pp. 2308–2323, July 2014.

[22] N. Fernández, R. J. Durán, I. de Miguel, N. Merayo, P. Fernández, J. C. Aguado, R. M. Lorenzo, E. J. Abril, E. Palkopoulou, and I. Tomkos, “Virtual topology design and reconfiguration using cognition: Performance evaluation in case of failure,” in *Proc. 5th Int. Congr. on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, Sept. 10–13, 2013, pp. 132–139.

[23] J. D. Cryer and K.-S. Chan, *Time Series Analysis: With Applications in R*, 2nd ed. Springer, 2010.

[24] J. L. Vizcaino, Y. Ye, F. Jiménez, R. Duque, and R. M. Krummrich, “Cost evaluation for flexible-grid optical networks,” in *IEEE Globecom Workshops*, Dec. 3–7, 2012, pp. 358–363.

[25] A. MacKenzie, J. H. Reed, P. Athanas, C. W. Bostian, R. M. Buehrer, L. DaSilva, S. W. Ellingson, Y. T. Hou, M. Hsiao, J.-M. Park, C. Patterson, S. Raman, and C. da Silva, “Cognitive radio and networking research at Virginia Tech,” *Proc. IEEE*, vol. 97, no. 4, pp. 660–688, Apr. 2009.

[26] T. Jiménez, J. C. Aguado, I. de Miguel, R. J. Durán, M. Angelou, N. Merayo, P. Fernández, R. M. Lorenzo, I. Tomkos, and E. J. Abril, “A cognitive quality of transmission estimator for core optical networks,” *J. Lightwave Technol.*, vol. 31, no. 6, pp. 942–951, Mar. 2013.

[27] N. Fernández, R. J. Durán, I. de Miguel, J. C. Aguado, T. Jiménez, M. Angelou, D. Sánchez, P. Fernández, N. Merayo, N. Atallah, R. M. Lorenzo, I. Tomkos, and E. J. Abril, “Cognitive genetic algorithms to design impairment aware virtual topologies in optical networks,” in *Proc. OFC*, 2012, paper OW3A.1.

[28] Dragon GMPLS website [Online]. Available: https://wiki.maxgigapop.net/twiki/bin/view/DRAGON.

[29] GNU Zebra website [Online]. Available: http://www.gnu.org/software/zebra.