MEICAN: Simplifying DCN Life-Cycle Management from End-User and Operator Perspectives in Inter-Domain Environments

Juliano Araujo Wickboldt, Maurício Quatrin Guerreiro, Lisandro Zambenedetti Granville, Luciano Paschoal Gaspary, Marcos Felipe Schwarz, Chin Guok, Vangelis Chaniotakis, Andrew Lake, and John MacAuley

The authors introduce MEICAN, a platform to manage the life-cycle of DCN from definition to provisioning of virtual circuits. They also present a case study of inter-domain circuit reservation with mixed manual and automated policy checking, and discuss lessons learned, relevant challenges, and open issues still to be overcome.

Abstract

National research and education networks (NRENs), such as ESnet, GéANT, and RNP, currently promote the employment of dynamic circuit networks (DCNs) to improve scientific communications beyond the capabilities of today's Internet. In spite of their alleged benefits to materialize user-initiated, ad hoc dedicated end-to-end circuits for high-demanding applications, DCNs also pose important challenges. Two examples are dealing with the end-user's lack of ability or willingness to understand low-level technicalities of virtual circuit establishment, and accommodating NRENs' local policies throughout the life-cycle of DCN. In this article, we seek to improve the usability of DCN services with a focus on the inexperienced end-user and the skilled network operator alike. We introduce MEICAN, a platform to manage the life-cycle of DCN from definition to provisioning of virtual circuits. We also present a case study of inter-domain circuit reservation with mixed manual and automated policy checking, and discuss lessons learned, relevant challenges, and open issues still to be overcome.

Introduction

Collaboration among researchers around the world is essential for the progress of science. This collaboration often involves running data-intensive e-science applications that are, in most cases, service-sensitive and have network requirements that cannot be met by the current best-effort Internet [1, 2]. To improve scientific communications beyond the capabilities of the Internet, initiatives such as the Global Lambda Integrated Facility (GLIF) promote the employment of dynamic circuit networks (DCNs) over national research and education networks (NREN), such as ESnet, GéANT, JGN-X, and RNP [3].

DCNs combine traditional packet-switched communication over IP with the predictability of circuit-switched networks to materialize user-initiated, ad hoc dedicated end-to-end circuits for high-demanding and real-time e-science applications. Despite their benefits, DCNs also pose important challenges. DCN end-users (e.g., researchers, students, and companies), for example, expect to establish circuits effortlessly, because the end-user is not usually willing to deal with the underlying technicalities (e.g., establishing VLANs, configuring MPLS) involved in stitching a circuit that traverses multiple national networks.

Even though NRENs participating in DCN-based collaborative environments are federated and thus agree on a common set of principles for providing DCN services, they still want local policies to be enforced and local requirements respected [4, 5]. For example, when a researcher connected to RNP (Brazil) needs to send experimentation data to a research facility connected to SURFnet (Netherlands), it is very likely that because of the current topology of transatlantic lines, the corresponding traffic will pass through at least one transit connection in the USA. In this case, not only the source and destination partners (RNP and SURFnet) need to agree that this service can be provisioned over the federated network, but also all intermediate networks need to allow the transit connection. Coordinating distributed circuit reservation and accommodating individual policies can be challenging, and in general it involves manual labor such as ticket tracking, email exchanges, and phone calls [6].

In this article, we address the issue of improving the usability of DCNs from the point-of-view of both unskilled end-users and network operators. We observe how end-users can express virtual circuit requirements in a friendly way. We also tackle the question of empowering operators to materialize circuit requests in the federated environment, while still preserving local policies. To conduct our observations, we present the design, implementation, and deployment of MEICAN, a Management Environment of Inter-domain Circuits for Advanced Networks. MEICAN is an open-source platform that enables end-users to request circuits over DCN-based networks. In addition, through the use of policy workflows, MEICAN facilitates collaboration among network operators of independent network domains, and leverages automation to speed up the inter-domain circuit reservation process. Based on open standards, such as the Network Service Interface (NSI) [7], MEICAN is able to interact with net-

Juliano Araujo Wickboldt, Maurício Quatrin. Guerreiro, Lisandro Zambenedetti Granville, and Luciano Paschoal Gaspary are with the Federal University of Rio Grande do Sul (UFRGS); Marcos Felipe Schwarz is with the National Research and Education Network (RNP); Chin Guok, Vangelis Chaniotakis, Andrew Lake, and John MacAuley are with the Energy Sciences Network (ESnet).

Digital Object Identifier: 10.1109/MCOM.2017.1601205

Currently, MEICAN still employs policy workflows. However, there are many improvements in both the design and execution of policies and the inter-domain circuit life-cycle management as a whole, leveraging NSI as a default interface.

work middleware to establish end-to-end QoS-enabled virtual circuits.

The remainder of this article is organized as follows. In the next section, we review the background of dynamic circuit reservation and associated network tools. Afterward, we present our approach to tackle current challenges, introducing MEICAN for inter-domain circuit reservation and detailing its key automations in DCNs. We then present a case study of inter-domain reservation with mixed manual and automated actions and discuss lessons learned in the process, relevant challenges, and open issues. Finally, we conclude the article with final remarks.

Background

After over a decade of research and development, several parallel efforts have been carried out to establish a global inter-domain DCN environment, culminating in the development of platforms that automate configuration, monitoring, and management tasks in this context. The On-Demand Secure Circuits and Advance Reservation System (OSCARS — https://www.es.net/engineering-services/oscars/) is one of these tools, which was developed as part of a U.S. Department of Energy research project and is currently used by ESnet, RNP, and other NRENs. Similar to OSCARS, the Automated Bandwidth Allocation across Heterogeneous Networks (AutoBAHN) — https://forge.geant.net/forge/display/AutoBAHN) is a virtual-circuit management platform mainly used at GéANT's pan-European backbone. For monitoring and performance evaluation, the Performance Focused Service Oriented Network Monitoring Architecture (perfSONAR) has been developed in a collaboration initiative among Internet2, ESnet, RNP, and GéANT [8].

In particular, OSCARS was conceived in response to findings from a science workshop report [9], which stated that (end-to-end) guaranteed bandwidth virtual circuits were one of the most important new network services needed. OSCARS is logically comprised of four functions [10]: a coordinator, which orchestrates the reservation and provisioning workflows; a resource manager, which tracks committed resources; a path-computation-engine, which determines routes; and a network element device driver, which communicates and configures network elements via the CLI or programmatic interface (e.g., NETCONF/YANG) to setup/teardown virtual circuits within the data plane. A key aspect of OSCARS is its ability to perform inter-domain end-to-end provisioning. This is done primarily through a programmatic interface designed and specified by the Open Grid Forum's (OGF) NSI working group (https://redmine.ogf.org/projects/nsi-wg).

NSI is a suite of Web services-based protocols using a software-defined networking (SDN) approach to enable multi-domain service delivery and management [7, 11]. Currently, the NSI framework defines specifications for three abstract services. The connection service specification describes the protocol and state machines used to deliver a network connection and models its life-cycle. The discovery service specification defines the protocol and data model used to describe meta-data associated with an NSI deployment. The topology service specification provides an abstraction of the physical network topology for use in connection requests and path finding.

NSI is built around the concept of software agents, referred to as network service agents (NSA), that interact with the network through a set of service primitives. An example of these interactions among NSI software agents in an inter-domain environment is depicted in Fig. 1. The ultimate requester agent (uRA) is the NSA that originates a service request and is typically associated with the end-user process. The ultimate provider agent (uPA) is associated with one or more networks and service requests for connections on those networks, coordinating with the local network resource manager (NRM), which, in turn, manages the network resources (e.g., OSCARS and AutoBAHN). An aggregator NSA is an orchestration engine for service requests received from uRAs and acts as a gateway to other NSAs within the network. An aggregator NSA anchors original connection requests, performs path finding, segments connection requests if needed, and orchestrates connection reservations with children NSAs. It is important to note that although NSI is the common interface for inter-domain service delivery and management, each domain participating in the end-to-end circuit provisioning environment is free to choose how it implements its own circuit segment (e.g., EoMPLS, PBB-TE, etc.). However, the circuit service instantiation as viewed from outside each individual domain must be consistent (e.g., Ethernet VLAN Tagged Service) to facilitate "stitching" of domain segments into an end-to-end circuit.

To facilitate the management of DCNs, Santanna *et al.*, [6] presented an approach based on business process management (BPM) methods to describe policies for authorizing virtual circuit reservation requests. The focus of that investigation was on improving human cooperation to speed up the process of manual evaluation and authorization of virtual circuit requests that passed through many independent network domains. The proposed solution was implemented based on the use of Web Service Business Process Execution Language (WS-BPEL) workflows relying on a specific execution engine and was incorporated into an earlier version of MEICAN. Currently, MEICAN still employs policy workflows. However, there are many improvements in both the design and execution of policies and the inter-domain circuit life-cycle management as a whole, leveraging NSI as a default interface.

MEICAN: A Management Environment of Inter-domain Circuits for Advanced Networks

In this section, we introduce our approach to simplify and automate DCNs, describing MEICAN's architectural organization as well as the interactions between MEICAN and other systems to establish inter-domain virtual circuits. As depicted in Fig. 1, MEICAN interacts with one or more aggregator NSAs using the NSI protocol for both requesting inter-domain circuits and gathering information and updates

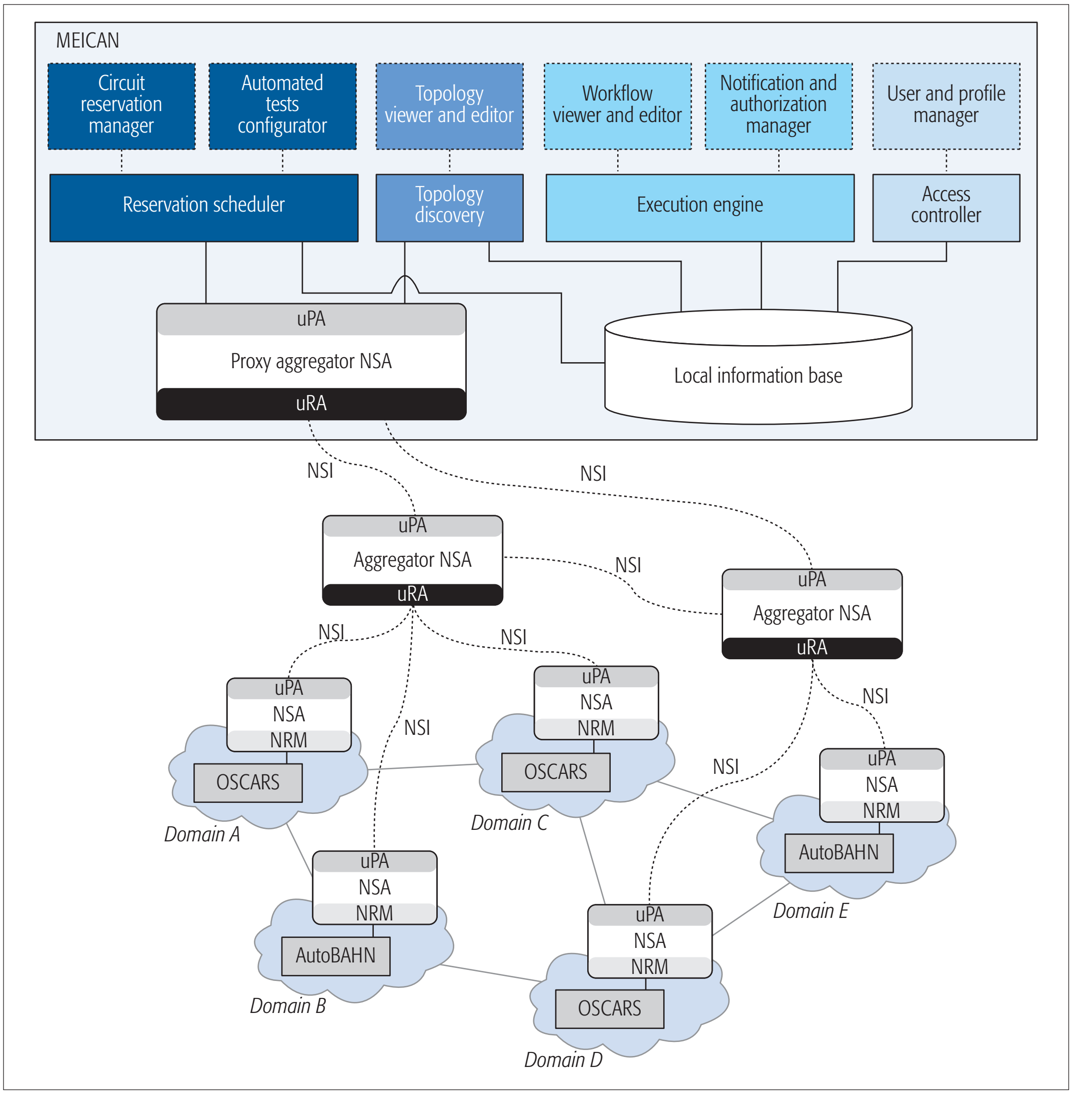

Figure 1. Main conceptual components of MEICAN interacting with the underlying NSI-based virtual circuit management infrastructure.

Although the technology to configure and establish inter-domain dynamic virtual circuits exists, one major hurdle is that the end-user, who can benefit from this service, is either incapable or unwilling to understand low-level protocol parameters. Therefore, one of the most fundamental functions of MEICAN is to provide an intuitive mechanism for the end-user to specify virtual circuit parameters.

of the underlying network infrastructure. Internally, MEICAN has six interface components (dashed-lined boxes), which serve the purpose of managing inter-domain circuit reservations, visualizing and troubleshooting the underlying topology, and controlling user accounts and profiles. In addition, four core components (solid-lined boxes) execute the commands coming from interfaces, and also handle events originated in the network infrastructure. Information on topology, circuit reservation requests, users, and profiles are all persisted in a *local information base*. Moreover, to participate in the NSI-based multi-domain global network federation, MEICAN implements a specific proxy aggregator NSA, which is used to receive topology updates and intermediate circuit requests coming from core components.

In the following subsections we highlight three of the most valuable and distinctive features of MEICAN. First, we describe how inter-domain circuit reservations are performed by the end-user. Then, the definition of authorization policies as workflows is detailed, and finally, we discuss how MEICAN is able to automate some periodic checks for configuration consistency.

Inter-Domain Circuit Reservation

Although the technology to configure and establish inter-domain dynamic virtual circuits exists, one major hurdle is that the end-user, who can benefit from this service, is either incapable or unwilling to understand low-level protocol parameters. Therefore, one of the most fundamental functions of MEICAN is to provide, through its circuit reservation manager component, an intuitive mechanism for the end-user to specify virtual circuit parameters. With a map-aided reservation mechanism (Fig. 2), the end-user is able to find service termination points (STPs) to specify a circuit's source and destination, in addition to informing the bandwidth required. Triangles on the map identify STPs (different colors represent different domains) that the user can select as source or destination of a circuit with a mouse click or using textual search. It is also possible to schedule recurrent virtual circuits (e.g., establish the same virtual circuit every Thursday at a given time of day). In addition, for advanced users, it is possible to specify waypoints over which the circuit needs to be routed. In the following section we provide further details on MEICAN's four-step circuit reservation wizard in a case study.

As soon as an end-user submits a new circuit

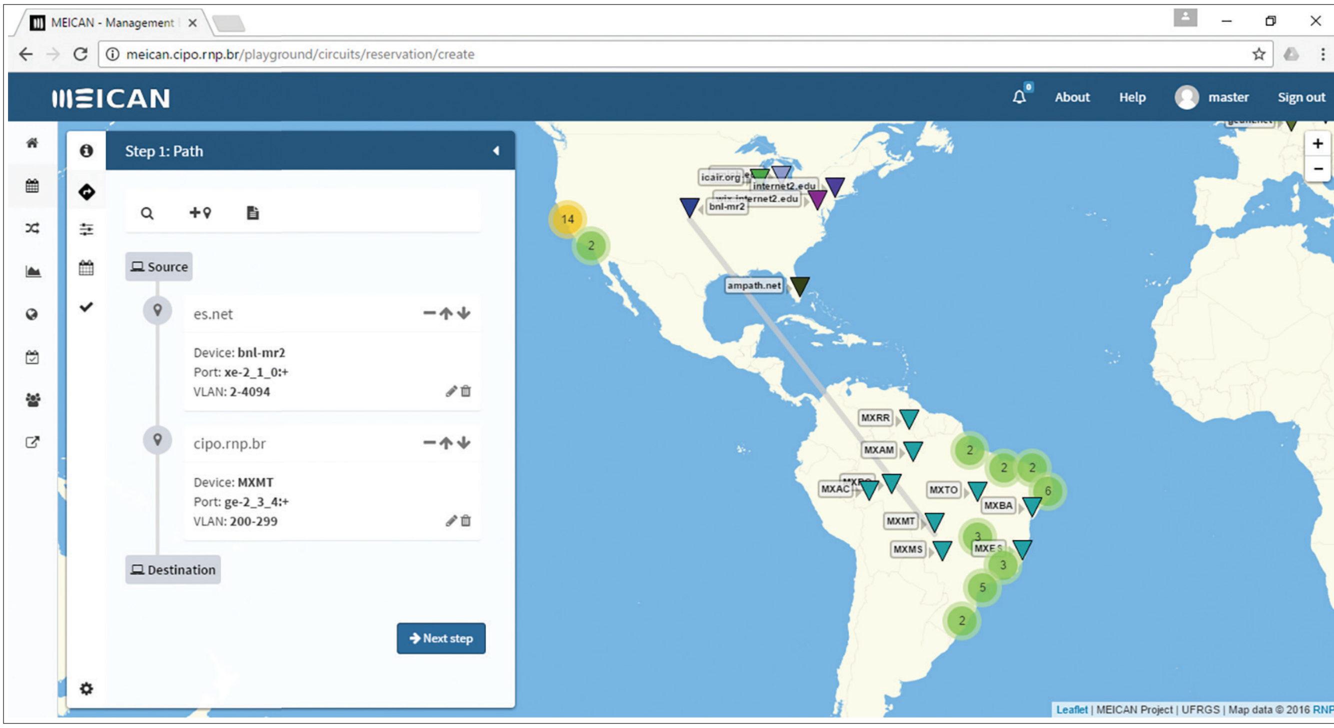

Figure 2. Screenshot of map-aided circuit reservation interface.

request, the *reservation scheduler* component comes into play to manage the whole life-cycle of the reservation. Figure 3 provides details on the system interactions and status changes coordinated by MEICAN during the life-cycle of a reservation request. MEICAN maintains three state machines, for reservation, authorization, and connectivity status. To interact with remote systems, MEICAN communicates using NSI messages via its local *proxy aggregator NSA*, which in turn redirects the messages created by MEICAN's internal components to the appropriate remote aggregator NSA. It is important to notice that for the sake of clarity, only success states and interactions are displayed in Fig. 3.

As a first step to fulfill a virtual circuit request, MEICAN, by means of the proxy aggregator NSA, submits a new reservation request to a given remote aggregator NSA system and receives back an identifier (*id*), which is used in subsequent message exchanges. Next, the remote aggregator NSA asynchronously replies, informing that resources are available for that request (e.g., a network path with sufficient bandwidth is available). MEICAN then commits the reservation and queries the aggregator NSA to discover the full path of the circuit to be established. Based on this path, MEICAN executes an internal authorization process running authorization workflows for each domain in the path. This process might include several automated and manual authorization steps (further detailed below). Once the reservation is authorized, MEICAN passes to a provisioning phase in which the virtual circuit is configured/scheduled in the network. Upon the scheduled activation time of the circuit, the remote aggregator NSA will notify MEICAN asynchronously once the circuit/dataplane state changes to *up*, causing MEICAN to update its local connectivity status (for the virtual circuit). Finally, when the termination time of the circuit is reached, the aggregator NSA will tear down the circuit (releasing the resources) and then inform MEICAN with another state change. MEICAN will record the new dataplane state and update its internal reservation state, keeping the information about the request for future reference.

Policy Workflows for Distributed Reservation Authorization

Another significant challenge in the context of dynamic inter-domain circuit reservation is respecting the independent policies that each participating domain wants to enforce locally. Let's assume that a circuit, requested from domain A toward domain C, may need to pass through another domain, B. In this case, all three domains' policies need to be observed before the request can be approved. Internally, each independent domain manages its own network with an NRM system. To participate in the federated network, each domain agrees to run a local NSA uPA. It speaks a common language (i.e., NSI) to propagate intra-domain information into the inter-domain context. However, NSI was not designed to express policies to access resources available in each domain.

MEICAN fills this gap through its *workflow viewer & editor* component, which allows domain operators to specify policies through a service-oriented workflow language. Via a drag-and-drop mechanism, as depicted in Fig. 4, operators choose the set of visual elements that leverage information of the circuit being requested, so to automate authorization decisions. Elements are of three types:

- Terminals, which specify either a starting point (i.e., arrival of a request) or ending of a workflow branch with a final authorization decision.

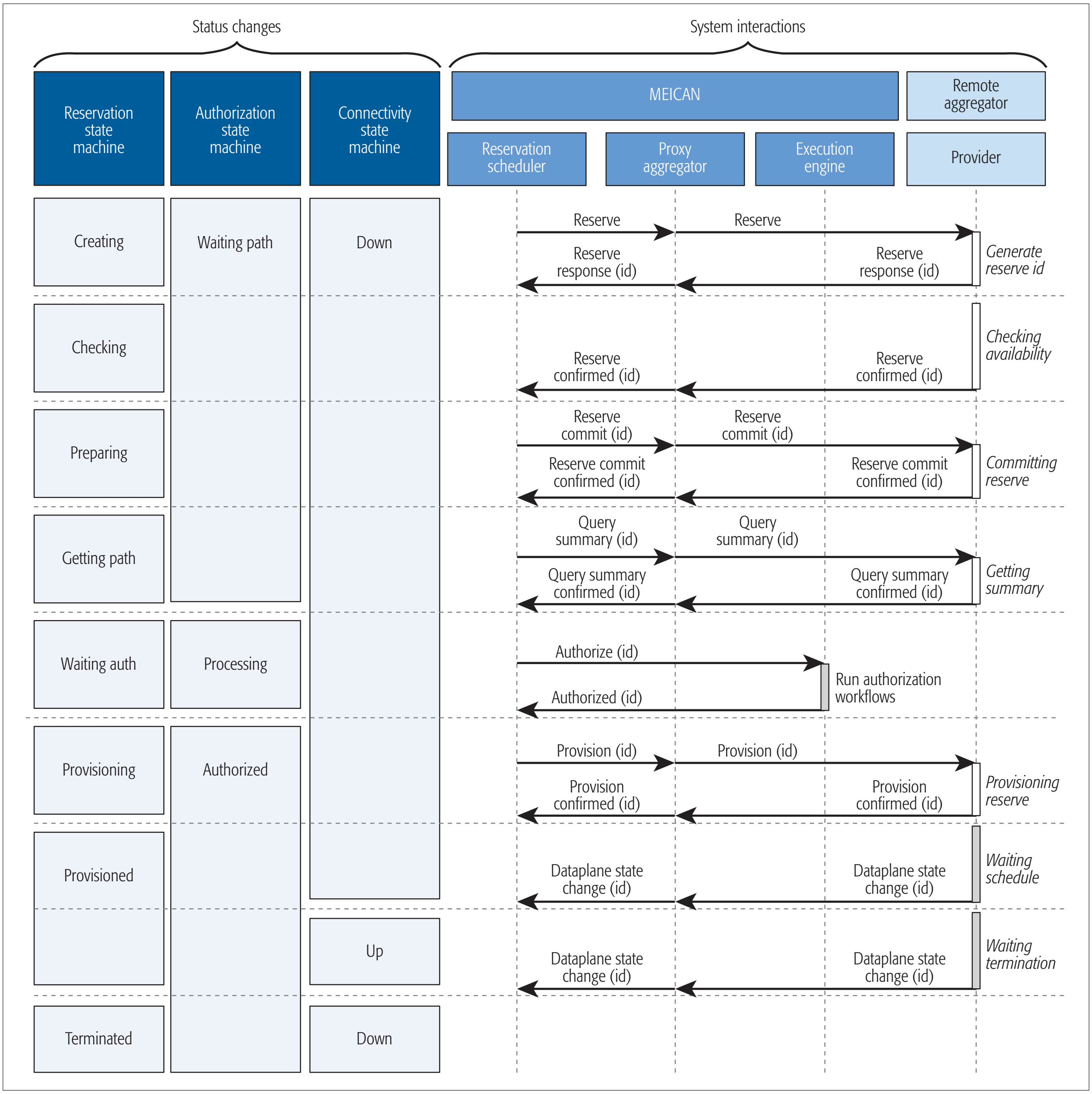

Figure 3. State control and system interactions throughout the life-cycle of virtual circuit reservations.

- Automation filters, which can make decisions based on information provided by a service (e.g., the user requesting a circuit or the total bandwidth requested).
- Operator intervention requests, which are employed in situations where not enough information is available for an automated decision, in which case domain operators will be notified to evaluate the virtual circuit request at hand and decide upon its authorization.

Moreover, it is important to emphasize that the visual workflow language employed by MEICAN is extensible by the addition of new elements. Since this language takes a service-oriented approach, virtually any information that can be obtained as a Web service (e.g., resource usage of a specific device) can be used to express more sophisticated local reservation policies.

For every virtual circuit request placed by users, MEICAN analyzes the request and the involved domains to choose which policy workflows need to be executed. The *execution engine* component is responsible for selecting the appropriate workflows accordingly and orchestrating a series of Web service calls until it reaches a final decision either to approve or deny a given request. Every time a manual authorization is required by a policy workflow, the *execution engine* halts the process, notifies the appropriate operators, and waits for a decision before proceeding. The *notification & authorization manager* component provides the appropriate mechanisms to notify domain operators, displays relevant information about the

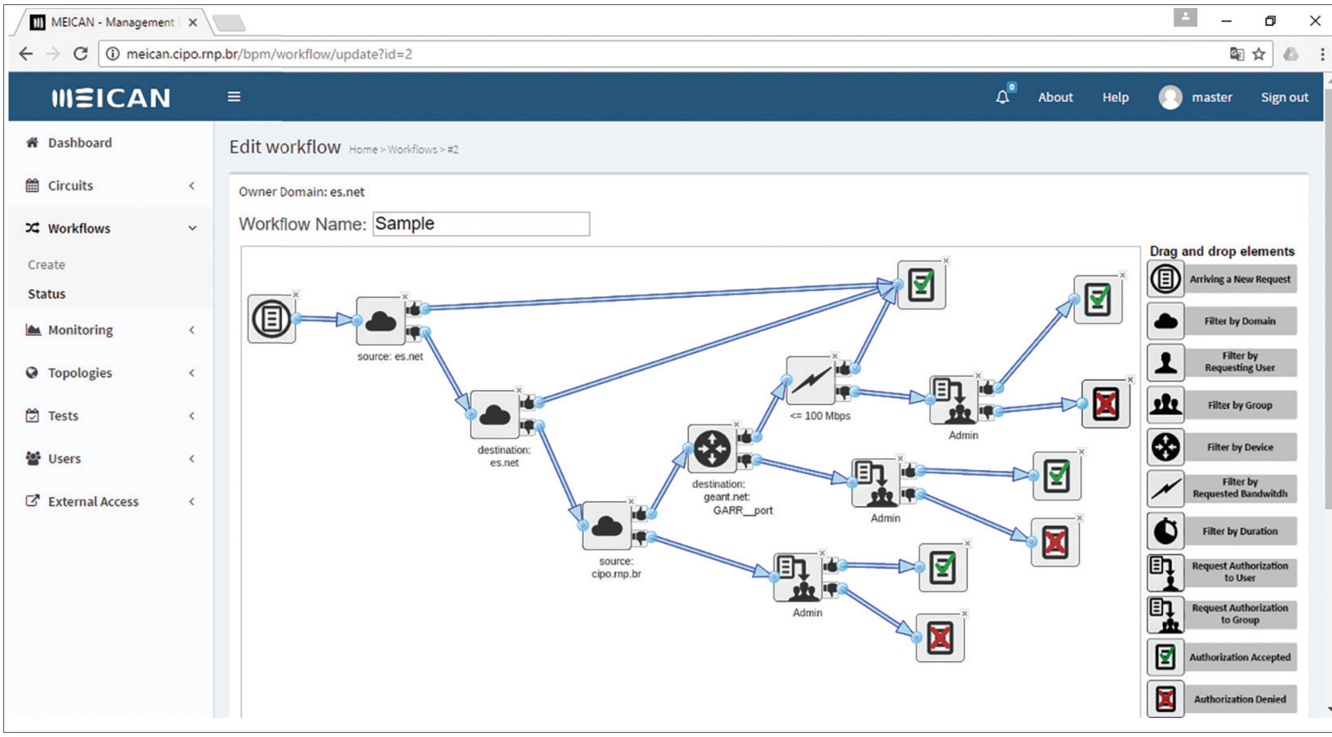

Figure 4. Screenshot of workflow editor for domain policy specification (the usage of this sample workflow is detailed in our case study.

request (e.g., full path, user details, bandwidth requested) to support decision making, and offers options to authorize or deny a request. When recurrent circuits are requested (e.g., daily, weekly, monthly requests), each recurrence may traverse a different set of intermediate domains, thus policy workflows are executed independently for each recurrence. Nevertheless, MEICAN groups recurrent authorization requests generating only one notification per request, allowing operators to authorize or deny each recurrent circuit individually or as a bundle.

Automation of Routine Consistency Check Tests

MEICAN operates over global scale multi-domain networks, an environment in which unadvertised changes occur very often (e.g., equipment upgrades, configuration changes, device renaming). Such changes need to be propagated and quickly reflected into the internal representation of all systems involved in the virtual circuit reservation and provisioning process. It is likely that MEICAN's *local information base* and aggregator's internal topology representations will eventually get out of sync and occasionally become inconsistent with the current network configuration. The issue in this context is that virtual circuit requests fail when the actual network configuration is inconsistent with the parameters sent over NSI by MEICAN and other systems. In general, network operators discover these inconsistencies only when users complain about failing reservation requests.

MEICAN's *automated tests configurator* component is able to simulate the user's behavior by placing reservation requests between network endpoints and evaluating the output status. This can be useful to figure out beforehand when a reservation would fail because a path cannot be found or a provider is temporarily offline. These automated tests can be configured to run periodically and the operator can choose which endpoints they want the test to run against. During these tests, MEICAN bypasses any authorization workflows to avoid stumbling upon authorization issues that have nothing to do with network configuration inconsistencies.

The *topology discovery* component of MEICAN, in turn, acts proactively trying to detect and resolve network configuration inconsistencies updating the *local information base*. There are three possible operation modes for this component:

- Manual, in which an operator manually starts the discovery process and verifies the report of changes resolving occasional inconsistencies.
- Automated by a scheduler, in which an operator schedules the discovery process to run and gets a notification whenever changes or inconsistencies that require human intervention occur.
- Automated by NSI notifications, in which MEICAN's proxy aggregator NSA subscribes to receive notification of topology changes from other *aggregator NSAs*.

In any of the operation modes, MEICAN is able to gather and process information coming from many sources and discover, for instance, newly added ports, updated endpoint/waypoint properties such as available VLAN ranges, and notify operators accordingly.

Case Study

To effectively provide user-driven inter-domain circuits, MEICAN must manage interactions with both end-users and network operators. The end-user is primarily concerned with how to

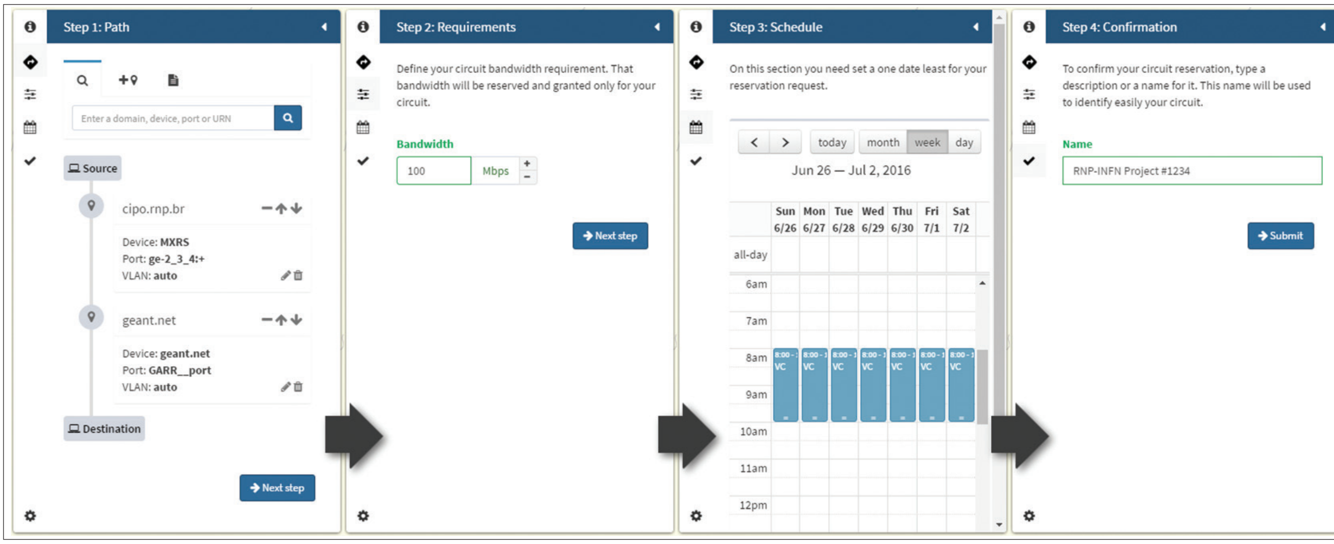

Figure 5. Steps to place a dynamic circuit reservation.

reserve an end-to-end circuit (e.g., selecting end-points, bandwidth, start/end time), the life-cycle status of the reservation (e.g., pending, committed, authorized), and the status of the provisioned path (e.g., up, down, failed). The operator, on the other hand, has a much larger scope of concerns, which include ensuring the topology is correct, policy workflows are accurately modeled, and information is available to audit reservations (e.g., status, pending manual authorizations, resource consumption). In this section, we present a case study on how MEICAN is able to facilitate this process from both the end-user's and operator's perspective.

Consider a specific situation in which a researcher at a university in Brazil connected to RNP needs to pull large datasets from the LHC Open Network Environment (LHCONE) located at a tier 1 site in the Istituto Nazionale di Fisica Nucleare (INFN) in Italy. More specifically, suppose that a new dataset will be generated to be collected every morning during an experimentation window that lasts for three months, and that this dataset can be transmitted over a 100 Mb/s dedicated connection safely within two hours. There is no direct connectivity between RNP and LHCONE, in a situation where an agreement between the two end-points would not be sufficient to establish this circuit. In that case, some intermediary domains are also needed to agree in allowing the circuit to transit over their backbones.

From the end-user's perspective, they need only to login to MEICAN's web interface — Brazilian researchers do this via CAFe federation (https://portal.rnp.br/web/servicos/cafe-en) — and follow the four steps of the circuit reservation wizard, as depicted in Fig. 5:

- Selecting source and destination using the search box or aided by the map view shown in Fig. 2.
- Specifying the required bandwidth.
- Scheduling the days and duration of the reservation in the calendar.
- Providing a name to identify the request.

Note that the end-user does not need to be aware that source (*cipo.rnp.br*) and destination (*geant.net*) are not directly connected, let alone knowing that some intermediary domain provides transit for the circuit.

After the request is placed by the end-user, let's assume that the aggregator NSA decides to select a path that transits through ESnet's backbone. As a general rule, ESnet has an acceptable use policy (AUP) that requires one end of the traffic, source or sink, to be its customer. In the specific request considered in this case study, neither source (RNP) nor destination (GéANT) are located within ESnet's domain, which means that this is a transit request and would need to be examined by one of its local operators. In this situation, since GéANT is participating in LHCONE (which is helping to fund the transatlantic lines) and RNP is an ESnet partner in South America, the operator at ESnet can decide to add to the local policy workflow in MEICAN a specific automated decision flow to authorize this and any other future circuit requests without human intervention, as depicted in Fig. 4. In ESnet's workflow, for any transit request (i.e., when neither source nor destination are within the *es.net* domain), if the source is RNP (*cipo.rnp.br*) and the destination is a specific LHCONE device/port in GéANT (*GARR_port*), the request will be automatically authorized with bandwidth up to 100 Mb/s. Otherwise, ESnet's operators will be notified by MEICAN and will need to inspect and decide about each request.

As an exercise of contrast, consider a situation in which MEICAN would not be orchestrating this virtual circuit request. The end-user in Brazil would need to contact his/her local operator, who in turn would open a ticket within RNP specifying the circuit requirements. Then, RNP would need to contact the remote site to agree on the terms of the circuit being requested, and both would need to decide upon the transit for that connection as well. Subsequently, either one of the endpoints (RNP or GéANT) would need to contact the transit domain operator, who would evaluate their local AUP to determine whether this is an acceptable request or not. From the authors' experience, coordinating a multi-continent end-to-end circuit usually requires at least 10

Since MEICAN supports input from operators in authorizing requests, it would be interesting to add support for end-user interaction elements, for example, forms to gather additional information about a specific request. This could allow a more precise representation of some AUPs that require further authentication of end-users.

to 20 emails (getting the right approvals, selecting the correct VLANs, and so on), five to six revisions of a topology diagram, and usually three to four phone conferences to ensure that the circuit is up and tested accordingly. These cumbersome interactions can be eliminated if all the information required for decision making is centralized and automated as much as possible by MEICAN, as evidenced by this case study.

Lessons Learned, Challenges, and Open Issues

Along the seven years of research and development of MEICAN, our solution to simplify DCN life-cycle management, many lessons have been learned and challenges remain to be tackled. From past experience, we have learned that setting up virtual circuits based only on bandwidth, endpoints, and waypoints is not always expressive enough to represent real application requirements. Much value can be added to the service of inter-domain virtual circuit establishment if end-users were able to express application requirements such as multi-point connections and redundant paths [12]. These types of requirements could benefit, for example, bandwidth guaranteed video conferencing with multiple parties or mission-critical applications such as telesurgery. However, there is a trade-off between creating a mechanism that would be straightforward to understand and interact with, while expressive enough to represent all the requirements of virtual circuits. In addition to creating such a mechanism, support for these virtual circuit requirements will need to be added to the NSI protocol suite and underlying virtual circuit provisioning systems in order to fulfill them.

Regarding local domain policy representation, we have been employing workflow design and execution mechanisms as a tool to minimize operators' manual labor. Automation in this context brings agility to the whole virtual circuit authorization process, as exemplified in our case study. However, it also comes at the cost of reducing implicit human supervision, making the system more prone to abuse. When human operators need to interact and oversee the authorization process, they naturally try to compromise and negotiate the acceptable use of their premises. In the automated approach, however, any request that passes the automation filters will be given resources automatically. For example, some domains might automatically block every request, while others can become overloaded because their policy workflows are too permissive. Therefore, it is utterly important to ensure that workflows designed by operators semantically match the specified local or global AUP. Moreover, classical approaches from the policy-based network management field, such as policy conflict resolution and policy refinement, can be leveraged in this context to create fairness and prevent abuse [13].

As we receive feedback from domain operators we keep improving the workflow system to accommodate a broader set of meaningful automation filters and interaction elements. Since MEICAN supports input from operators in authorizing requests, it would be interesting to add support for end-user interaction elements, for example, forms to gather additional information about a specific request (e.g., authorization token or key). This could allow a more precise representation of some AUPs that require further authentication of end-users. To minimize manual labor even further, learning methods could be used to remember manual authorization decisions. In addition, workflow granularity could be changed from one workflow per domain to a per device or per port basis to create simpler workflows that execute for requests in specific regions of the a local domain. Finally, MEICAN's workflow execution engine can also be employed to provide "authorization as a service" when circuit requests are placed by other applications that rely on DCN services (e.g., ATER [14]). In this case, MEICAN could become the bond of trust among DCN service providers, DCN-based applications, and end-users [5].

On consistency checking, from the point-of-view of the operator, a current open issue is that MEICAN's automated testing procedures only cover the reservation system (e.g., communication with uPAs to place a circuit request and obtain a valid path); neither circuit configuration nor provisioning procedures are tested. To verify the correct behavior of all components involved throughout the whole DCN life-cycle, some level of integration with monitoring and performance checking systems could be included (e.g., using tools such as ESnet Monitoring Daemon (http://software.es.net/esmond/) or CMon [15] with perfSONAR-MDM to inject traffic into one end of a virtual circuit and verify it at the other end). In addition, in this massive collaboratively managed distributed system that is a global inter-domain environment, troubleshooting network misconfigurations and eventual inconsistencies can be very challenging. Since MEICAN already subscribes to network changes and automatically adapts its local information base, one possible way to aid troubleshooting is to keep track of network configuration changes over time. This would allow, for example, the correlation information of failing requests with changes made in network configuration in the past.

Conclusion

In this article, we have addressed the issue of improving the usability of DCN services over NRENs through MEICAN, an open-source (MEICAN's GitHub repository: https://github.com/ufrgs-hyman/meican) platform developed to manage the life-cycle of DCN from definition to provisioning of virtual circuits. MEICAN focuses both in the end-user who is unwilling to deal with networking technicalities (e.g., establishing VLANs, configuring MPLS), and also the experienced network operator who wants mainly to express and see enforced their local domain policies and keep the network configuration consistent. We have presented a case study of inter-domain (and inter-continental, that is, from the Americas to Europe) virtual circuit reservation, including mixed manual and automated policy enforcement decisions through our workflow system, emphasizing how MEICAN can be employed to minimize manual labor in this context. Finally, we have discussed some important challenges and issues that remain open in the context of DCN.

References

[1] N. Charbonneau *et al.*, "Advance Reservation Frameworks in Hybrid IP-WDM Networks," *IEEE Commun. Mag.*, vol. 49, no. 5, 2011, pp. 132–39.

[2] L. Zuo, M. M. Zhu, and C. Q. Wu, "Fast and Efficient Bandwidth Reservation Algorithms for Dynamic Network Provisioning," *J. Network and Systems Management*, vol. 23, no. 3, 2015, pp. 420–44.

[3] C. P. Guok *et al.*, "A User Driven Dynamic Circuit Network Implementation," *Proc. 3rd Int'l. Workshop Distributed Autonomous Network Management Systems (DANMS)*, IEEE GLOBECOM Workshops, IEEE, 2008, pp. 1–5.

[4] I. Monga *et al.*, "Hybrid Networks: Lessons Learned and Future Challenges Based on ESnet4 Experience," *IEEE Commun. Mag.*, vol. 49, no. 5, 2011, pp. 114–21.

[5] L. Gommans *et al.*, "The Service Provider Group Framework: A Framework for Arranging Trust and Power to Facilitate Authorization of Network Services," *Future Generation Computer Systems*, vol. 45, 2015, pp. 176–92.

[6] J. Santanna, J. Wickboldt, and L. Granville, "A BPM-Based Solution for Inter-Domain Circuit Management," *Proc. IEEE Network Operations and Management Symposium (NOMS)*, Maui, USA, 2012, pp. 385–92.

[7] T. Kudoh, G. Roberts, and I. Monga, "Network Services Interface: An Interface for Requesting Dynamic Inter-Data-Center Networks," *Proc. Optical Fiber Commun. Conf. Exposition and the National Fiber Optic Engineers Conf. (OFC/NFOEC)*, Mar. 2013, pp. 1–3.

[8] V. Bajpai and J. Schönwälder, "A Survey on Internet Performance Measurement Platforms and Related Standardization Efforts," *IEEE Commun. Surveys Tutorials*, vol. 17, no. 3, thirdquarter 2015, pp. 1313–41.

[9] R. Bair *et al.*, "High-Performance Networks for High-Impact Science," Office of Science — U.S. Department of Energy, Washington, DC, USA, Report of the High-Performance Network Planning Workshop, Aug. 2002; available: http://www.es.net/assets/Uploads/Highperformancenetworks-report.pdf.

[10] D. Robertson *et al.*, "Intra and Interdomain Circuit Provisioning Using the OSCARS Reservation System," *Proc. 3rd Int'l Conf. Broadband Communications, Networks and Systems*, Oct. 2006, pp. 1–8.

[11] K. J. Kerpez *et al.*, "Software-Defined Access Networks," *IEEE Commun. Mag.*, vol. 52, no. 9, 2014, pp. 152–59.

[12] A. Mendiola *et al.*, "Multi-Domain Bandwidth on Demand Service Provisioning using SDN," *Proc. 2016 IEEE NetSoft Conf. Workshops (NetSoft)*, June 2016, pp. 353–54.

[13] A. Schaeffer-Filho, E. Lupu, and M. Sloman, "Federating Policy-Driven Autonomous Systems: Interaction Specification and Management Patterns," *J. Network and Systems Management*, vol. 23, no. 3, 2015, pp. 753–93.

[14] K. Cardoso *et al.*, "Using Traffic Filtering Rules and OpenFlow Devices for Transparent Flow Switching and Automatic Dynamic-Circuit Creation in Hybrid Networks," *J. Systems and Software*, vol. 117, 2016, pp. 113–28.

[15] H. Yu *et al.*, "GéANT perfSONAR MDM-Based Circuit Monitoring in a Multidomain Environment," *IEEE Commun. Mag.*, vol. 52, no. 5, May 2014, pp. 174–81.

Biographies

Juliano Araujo Wickboldt (jwickboldt@inf.ufrgs.br) is a postdoctoral researcher at the Federal University of Rio Grande do Sul (UFRGS) in Brazil. He holds both M.Sc. (2010) and Ph.D. (2015) degrees in computer science from UFRGS. He was an intern at NEC Labs Europe in Heidelberg, Germany for one year between 2011 and 2012. In 2015, he was a visiting researcher at the Waterford Institute of Technology in Ireland. His research interests include softwarized networking and 5G technologies.

Maurício Quatrin Guerreiro (mqguerreiro@inf.ufrgs.br) is a software engineer. He holds a degree in computer engineering at the Federal University of Rio Grande do Sul (UFRGS) in Brazil. He currently works on projects focused on network management. During under graduation at UFRGS, he was a member of the Networks Group of the Institute of Informatics, where he participated in research involving partnerships with large companies and the Brazilian National Research and Educational Network (RNP).

Lisandro Zambenedetti Granville (granville@inf.ufrgs.br) received his M.Sc. (1998) and Ph.D. (2001) degrees in computer science from Federal University of Rio Grande do Sul (UFRGS). He is currently an associate professor with the Institute of Informatics of UFRGS. He is a member of the Brazilian Internet Committee (CGI.br), the Chair of the Committee on Network Operations and Management (CNOM) of IEEE ComSoc, and the co-chair of the Network Management Research Group (NMRG) of IRTF.

Luciano Paschoal Gaspary (paschoal@inf.ufrgs.br) holds a Ph.D. in computer science (UFRGS, 2002). He is deputy dean and associate professor at the Institute of Informatics–UFRGS, Brazil. He has been involved in various research areas, mainly computer networks, network management and computer system security. He is the author of more than 120 full papers published in leading peer-reviewed publications. In 2016, he was appointed as a Publications Committee member of the IEEE SDN initiative.

Marcos Felipe Schwarz (marcos.schwarz@rnp.br) has been a research & development coordinator at the Brazilian National Research and Education Network (RNP) since 2014. He received a B.Sc. degree in computer science from Santa Catarina State University (UDESC) in 2011, and an M.S. degree in computer engineering from the University of Sao Paulo (USP) in 2014. He works at RNP coordinating R&D projects involving advanced Internet, network performance monitoring, dynamic circuit networks, and he is involved in research and experimentation related to software-defined networking.

Chin Guok (chin@es.net) is the technical lead of the ESnet On-demand Secure Circuits and Advanced Reservation System (OSCARS) project, which enables end-users to provision guaranteed bandwidth virtual circuits within ESnet. He also serves as a co-chair of the Open Grid Forum On-Demand Infrastructure Service Provisioning Working Group. He received a M.S. in computer science from the University of Arizona in 1997 and a B.S. in computer science from the University of Pacific in 1991.

Vangelis Chaniotakis (haniotak@es.net) is a member of the Network Engineering Group and a core software developer for the ESnet OSCARS virtual circuit reservation and provisioning system. He is also involved in several international standardization activities related to virtual circuits, such as the OGF Network Standards Interface working group, and he is the chair of the GLIF GNI API Task Force. He was a network and software engineer at the University of Crete NOC, where he designed and implemented web-based systems and services.

Andrew Lake (andy@es.net) is a software engineer at ESnet. He has been a developer involved with the OSCARS project since 2005. He is also the lead developer and release manager for the perfSONAR project, a suite of open source tools for performing network monitoring and measurement deployed at thousands of locations around the globe.

John MacAuley (macauley@es.net) holds an M.Sc. in computer science from Western University, and is currently a researcher in the Office of the CTO at the Energy Sciences Network. He is a primary contributor of NSI standards at the Open Grid Forum and has developed production deployments of the NSI protocol for a number of networks. He has an extensive background in networking standards, software architecture, and software development at BNR, Nortel, Avaya, and SURFnet.

Since MEICAN already subscribes to network changes and automatically adapts its local information base, one possible way to aid troubleshooting is to keep track of network configuration changes over time. This would allow, for example, the correlation information of failing requests with changes made in network configuration in the past.