Service-Oriented Virtual Private Networks for Grid Applications

Hanxi Zhang*, Michel Savoie, Scott Campbell
Communications Research Centre Canada,
{hanxi.zhang, michel.savoie, scott.campbell}@crc.ca
Sergi Figuerola
i2CAT Foundation, Spain, sergi.figuerola@i2cat.net
Gregor von Bochmann
University of Ottawa, Canada, bochmann@site.uottawa.ca
Bill St. Arnaud
CANARIE Inc., bill.st.arnaud@canarie.ca

Abstract

Emerging Grid applications desire not only high bandwidth but also the ability to control the topology and traffic engineering of the underlying networks, through web service interfaces. To achieve that goal, we present an advanced User Controlled Lightpath Provisioning (UCLP) system, where network resources and Grid resources are both modeled as web services and are seamlessly integrated into workflows.

1. Introduction

The emerging data-intensive Grid applications desire networks capable of transferring bulk files in the order of at least Gigabytes per second [1,2]. Furthermore, many Grid applications need a persistent infrastructure for sustained data flows between instruments, computers and so forth. In many cases, those huge persistent data flows necessitate networks optimized for specific Grid applications.

User Controlled Lightpath Provisioning (UCLP) [3,4,5,6] has been proposed to solve the above problem. In the original UCLP systems, circuit-switched end-to-end (e2e) lightpaths are set up as the data plane for distributed Grid applications, bypassing the commodity Internet. Using the UCLP system, a human user or a Grid application can provision end-to-end (e2e) lightpaths on an inter-domain basis, without the aid of network administrators.

During the deployment and demonstrations of early versions of the UCLP software, it has been realized that the networking requirements of Grid applications go beyond provisioning an e2e lightpath between a pair of end points. We summarize two new challenges as follows.

First, UCLP should be generalized as a network provisioning and configuration tool, which partitions both switches and routers into virtual network resources. The control of these virtual network resources is then handed over to Grid users, so that they can create their own virtual private networks (VPNs), which have mesh topologies. Such VPNs can be either switched, routed, or a blend of both. The concept of network virtualization is seen in various recent research projects and initiatives [7], and the IETF also has an active work group on Layer 1 Virtual Private Networks (L1VPN) [8]. Using the UCLP software, a Grid user would be able to change the topology of his switched VPNs, so that he can control the connectivity between different sites in his Grid, to trial different Grid application scenarios. Similarly, in a routed VPN, a Grid user would be able to control routing tables and policies, and achieve desired traffic-engineering effects. To differentiate between these next-generation VPNs and their traditional counterparts, we refer to them as Articulated Private Networks (APNs) [9].

Second, it is a natural step to integrate virtual network resources in UCLP with non-network components of Grids, such as instruments, storage devices, computation farms, and visualization equipment, etc., into workflows. Grid applications nowadays are adopting the Service Oriented Architecture (SOA) and web service standards. In UCLP all virtual network resources will be modeled as web services. Using a workflow language such as the Business Process Execution Language (BPEL) [10, 11], we can integrate the network with Grid applications into web service workflows. A major benefit of using workflows is that, once deployed, a BPEL workflow becomes a web service in its own right, which can then be orchestrated into a higher-level workflow. That recursive approach would lead to high scalability and reusability.

There are a few on-going projects on re-configurable networks for Grid applications, such as DRAGON [12], CHEETAH [13], OSCARS [14], and DRAC [15], etc. Using these systems, dedicated e2e connections can be set up for Grid applications, and

some of these systems also adopt the SOA. What these systems have in common is that they provide bandwidth-on-demand and bandwidth reservation services, by using Generalized Multi-Protocol Label Switching (GMPLS) family of protocols as the inter-domain control plane. Inter-domain signaling and routing are intrinsically complex problems, and one has no choice but to deal with them if efficient use of bandwidth is desired. In contrast, UCLP is an intra-domain problem, because an APN's owner has the illusion that resources acquired from different domains collectively form a private network, which he has full control over. We acknowledge that by dedicating resources to specific users on a long-term basis, UCLP's management of bandwidth is not as fine-grained as bandwidth-on-demand systems. On the other hand, UCLP is ideal for those Grid applications that involve continuous large traffic flows between a set of known end points, among which topology and routing need to be changed from time to time.

We present in this paper a new UCLP system architecture based on web services and workflows, to reflect the evolution of UCLP philosophy, and address the above new challenges. In particular, with a workflow-oriented mindset, we model the majority of UCLP system services as BPEL workflows.

The rest of this paper is organized as follows: in Section 2 we introduce the overall UCLP system architecture; in Section 3 we discuss web service enabled virtual network resources; in Section 4 we present the modeling of BPEL workflows that link virtual network resources together; in Section 5 we conclude the paper.

2. System Architecture

The UCLP system architecture in terms of the software and service layers is shown in Fig. 1. In the Resource Management Layer, web service enabled virtual network resources are created from physical networks. The Service Orchestration Layer involves integrating virtual network resources into web service workflows, which assist users in carrying out network provisioning tasks. Finally, human users access underlying web services through graphical user interfaces (GUIs). Alternatively, those services may be accessed by applications through web service interfaces.

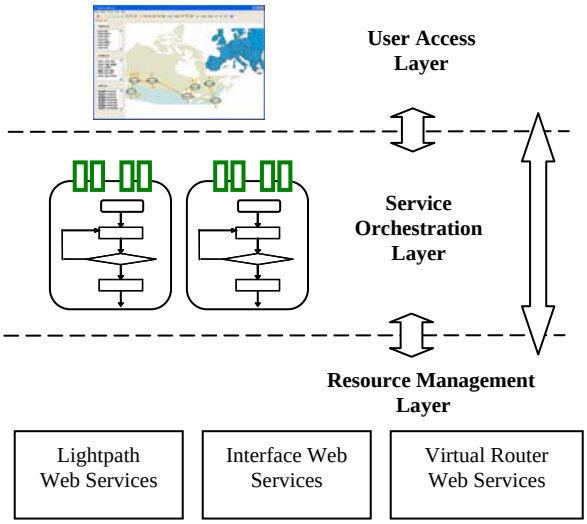

Figure 1 UCLP system architecture in three layers

In the UCLP system, we consider three types of virtual network resources: lightpaths, interfaces, and virtual routers, as shown in Fig. 2. A lightpath is a channel that connects two switches back-to-back, and based on the technologies of the switches, the channel could be a wavelength, a SONET/SDH channel, a VLAN, etc. An interface is an add/drop port on a switch, and is typically an Ethernet port. A virtual router is a partition of a physical router. Multiple virtual routers can be created on a physical router, each having its own dedicated ports, independent routing tables and policies [16].

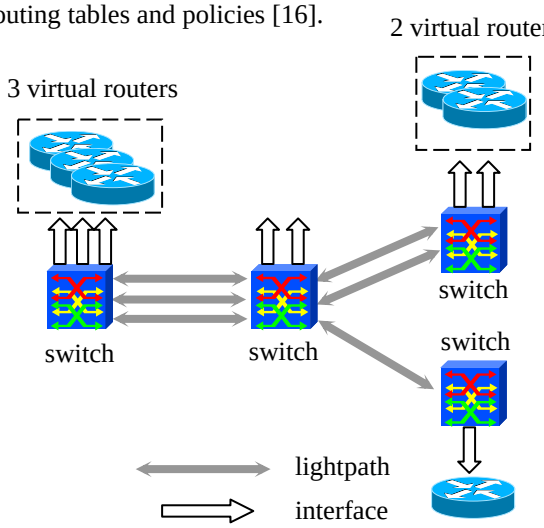

Figure. 2 Lightpaths, interfaces and virtual routers

Those three types of virtual network resources are fundamental building blocks of APNs, and they will all be made web services. Through web service interfaces, users can concatenate lightpaths to make a longer connection, unlink concatenated lightpaths from each other, partition a lightpath into lightpaths

with smaller bandwidth, bond lightpaths together to make a higher bandwidth lightpath, and sublease a lightpath to another user, etc. Interfaces can be concatenated with or unlinked from lightpaths, but cannot be partitioned or bonded. Virtual router web services will assist users in controlling routing tables and routing policies of virtual routers independently of each other.

The UCLP system architecture shown in Fig. 1 is from the perspective of a single Grid. A Grid could acquire virtual network resources from both physical networks and other Grids, in the form of APN resource lists. An APN resource list is an XML file containing a list of pointers to web services representing virtual network resources. Upon receiving an APN resource list from a provider (either a physical network or another Grid), a Grid has ownership over enclosed virtual network resources for an agreed-upon period. To achieve a loosely-coupled UCLP system, the transfer of APN resource list files is done via external means, such as email or secure file transfer, etc.

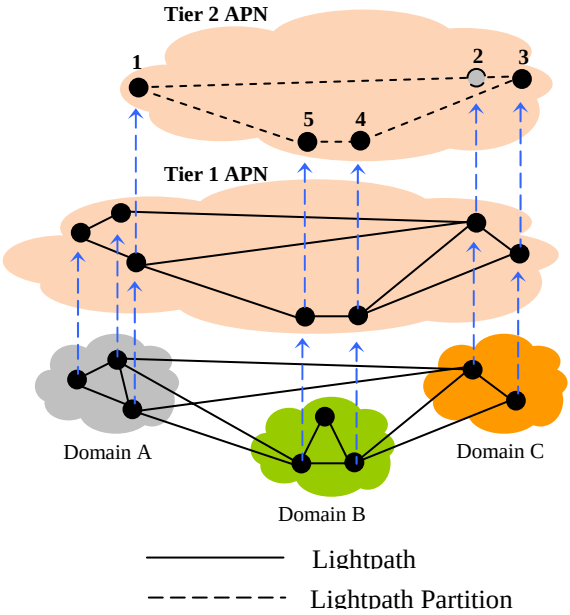

Figure 3 Recursive creation of APNs

When a Grid receives APN resource lists from different providers, the virtual network resources from all sources collectively form the Grid's APN. The boundaries between domains collapse, and a Grid ends up having its own private network, which may be geographically dispersed. The transfer of ownership over virtual network resources can be done in a recursive fashion. A Grid may choose to sublease some of its unused resources to another Grid. In Fig. 3, we illustrate the recursive creation of APNs. In Fig. 3, three physical networks make some of their resources available as lightpaths to a tier 1 APN, and the tier 1 APN partitions a portion of the lightpaths it owns, and subleases four lightpaths to a tier 2 APN. Note that the tier 1 APN concatenates lightpath partitions 1-2 and 2-3 into a new lightpath 1-3, and then gives it to the tier 2 APN. As a result, the tier 2 APN will only see lightpath 1-3, and switch 2 will become transparent.

3. Resource Management Layer

As of this writing, very few network element (NEs) supports on-board web-service-based management interface. Consequently, in the Resource Management Layer stand-alone controller web services are deployed. The switch-controller or router-controller web services communicate with the network elements using Transaction Language One (TL1), or Command Line Interface (CLI), etc.

In the Resource Management Layer, the creation and deletion of virtual network resources are done through factory services, which delegate to NE-controller web services. Each type of virtual network resource has its own factory service. Due to the fact that web service enabled virtual routers are still in a conceptual stage, in this paper we will focus our discussions on the lightpath factory service and lightpath web services. The interface factory service and interface web services will be very similar.

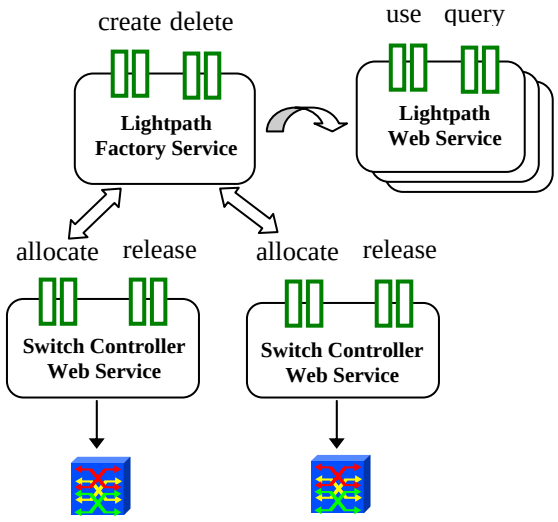

Figure 4 The lightpath factory service

The behaviors of the lightpath factory service are depicted in Fig. 4. When the lightpath factory service is requested to create a new lightpath web service, it signals the two switch-controller web services

corresponding to the end-points of the lightpath to allocate a slot, a port and a channel, and then deploys a new lightpath web service representing this channel. Similarly, when the lightpath factory service is requested to delete a lightpath web service, it signals the switch-controller web services at the end-points to release the corresponding slot, port and channel, and then terminates the lightpath web service.

Lightpath web services provide a layer of abstraction, by maintaining a set of generic state attributes, such as connectivity, bandwidth, and ownership, etc., and hiding technology-specific information from users. As users make use of lightpath web services, their states change accordingly. 'Use' and 'query' are a pair of operations by which the states of a lightpath web service can be modified and retrieved. Switch-controller web services are responsible for mapping lightpath web services to specific layer2 technologies.

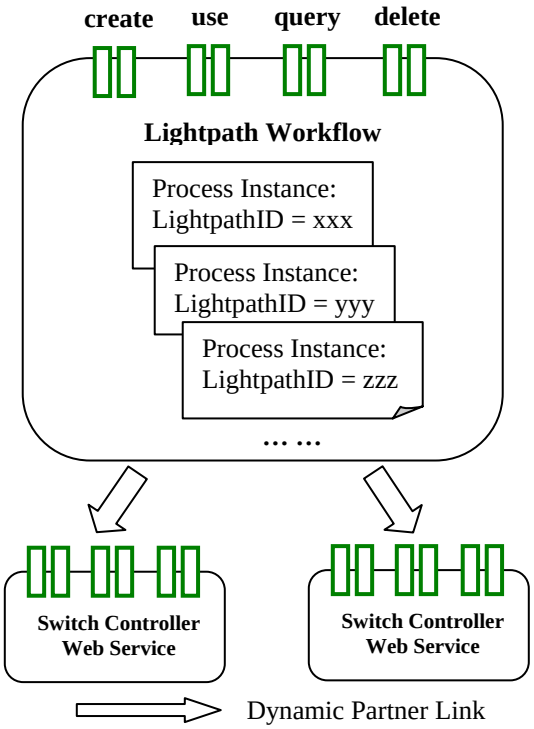

Figure 5 The lightpath workflow

When it comes to implementing the above factory pattern in the Resource Management Layer, we could implement both the lightpath factory service and lightpath web services as independent Axis [17] web services, each running at a unique Uniform Resource Identifier (URI). That approach does not scale well because it is possible but not convenient to manage web service life cycles through plain Axis, especially as the number of lightpath web services increases. In our solution, the lightpath factory service and lightpath web services are combined into what we refer to as the Lightpath Workflow, which is the orchestration of two switch-controller web services. The BPEL modeling of the Lightpath Workflow is shown in Fig. 5.

The Lightpath Workflow is a web service in its own right, and provides both factory operations ('create' and 'delete'), and state management operations ('use' and 'query'). The Lightpath Workflow has two switch-controller partner links, i.e., web services with which it interacts. Furthermore, those two partner links are dynamic and decided on-the-fly. When creating a lightpath, a user specifies the endpoint references to a pair of switch-controller web services, and those two dynamic partner links will be populated accordingly. In this way, a physical network only needs to run one single Lightpath Workflow, which dramatically simplifies the deployment of the UCLP system.

In a Lightpath Workflow, each lightpath is represented as a so-called process instance and is identified by a BPEL correlation tag. A BPEL process instance is like a thread that gets spawned by a workflow's start activity, in the case of the Lightpath Workflow, the 'create' operation. BPEL correlation is a mechanism by which a BPEL engine forwards an inbound SOAP message to the targeted process instance, so that each process instance has its own private conversation. In the Lightpath Workflow, the correlation tag is the lightpath's identifier.

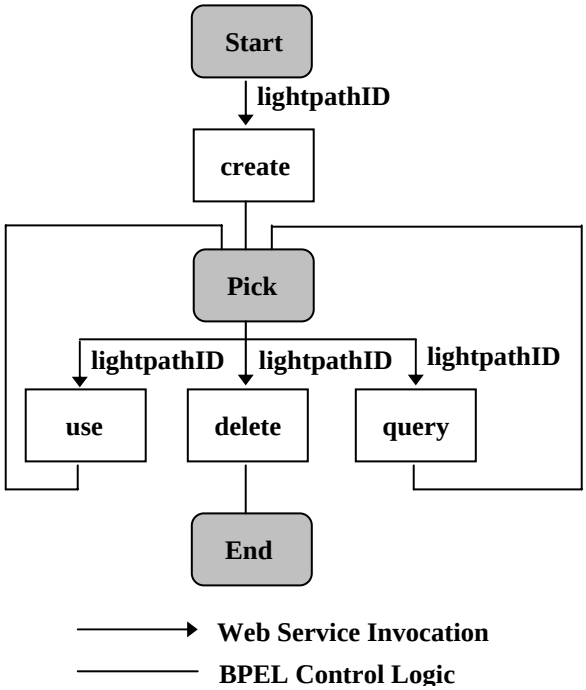

Figure 6 Life cycle of a lightpath

The combination of process instance and BPEL correlation techniques can be better understood by looking at the life cycle of a lightpath, as shown in Fig. 6. In Fig. 6 a white rectangle represents a Lightpath Workflow operation.

A lightpath is a long-running process instance that gets spawned by the 'create' operation. A user requesting to create a lightpath must specify a system-generated unique lightpath identifier (ID), which is to be matched by the 'use', 'query' and 'delete' operations with regard to this process instance.

Once created, a lightpath enters a pick block, where it accepts a 'use', a 'query', or a 'delete' operation. The lightpath will only exit the infinite pick block when the 'delete' operation is invoked, in which case the process instance terminates. A lightpath stores its states in a set of BPEL variables, and different lightpaths have different copies of those state variables. A lightpath's states first get initialized by the 'create' operation, and later get modified by the 'use' operation, which encompasses all possible usages on a lightpath, including: concatenate, unlink, partition, bond and sublease. From the authentication and authorization point of view, the 'create' and 'delete' operations are to be invoked by the provider of lightpaths and interfaces, while the 'use' and 'query' operations are to be invoked by the current owner of lightpaths and interfaces

We will have an Interface Workflow that works in the same fashion as the Lightpath Workflow, and also supports the same set of operations, except that it only has one dynamic partner link to a switch-controller web service.

In summary, the Lightpath Workflow solution offers a couple of benefits. First, it is much more lightweight and manageable then having all lightpaths as independent web services. Second, as long as the WSDL interface of the lightpath web service remains unchanged, one can modify the BPEL logics of the Lightpath Workflow and then re-deploy it, without affecting the Lightpath Workflow's client modules.

4. Service Orchestration Layer

The service orchestration layer is concerned with linking virtual network resource web services into higher-level BPEL workflows, thus providing users with an end-to-end solution. There are two types of BPEL workflows in the service orchestration layer: utility workflows and custom workflows.

When a user requests to perform day-to-day network provisioning tasks upon a set of selected resources, such requests will be handled by utility workflows. Utility workflows are specialized tool that come with the UCLP system, and currently we have implemented two of them: a ConnectionManagement Workflow that concatenates lightpaths and interfaces together, and a PartitionManagement Workflow that partitions a lightpath into children lightpaths or bonds children lightpaths back.

Utility workflows are intended for users who use the UCLP system as an immediate provisioning tool. In contrast, a custom workflow represents one or more network configuration scenarios tailored for a specific user. Using the UCLP system as an authoring tool, a user defines what needs to be provisioned in each scenario. As a result, a custom BPEL file is generated on-the-fly.

Fig. 7 illustrates the deployment of workflows in a physical network and in an APN. A physical network will run in its Tomcat application server a Lightpath Workflow and an Interface Workflow, which are for creating lightpath and interface resources from the physical network. An APN will run in its Tomcat application server a Lightpath Workflow, a set of utility workflows, and optionally custom workflows. The Lightpath Workflow at the APN level is for creating and managing derivative lightpaths, i.e., those created as a result of concatenating, partitioning or bonding acquired lightpaths. Custom workflows do not have to be deployed in the same BPEL engine where utility workflows reside. In fact, it is more likely that users deploy and use them outside the UCLP system.

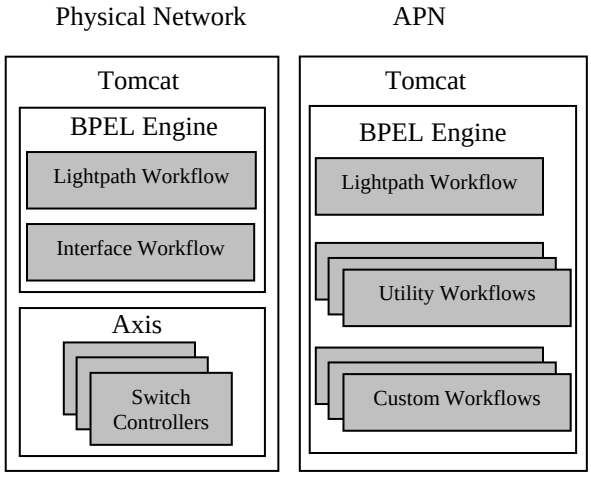

Figure 7 Workflow deployment in UCLP

4.1 Utility Workflows

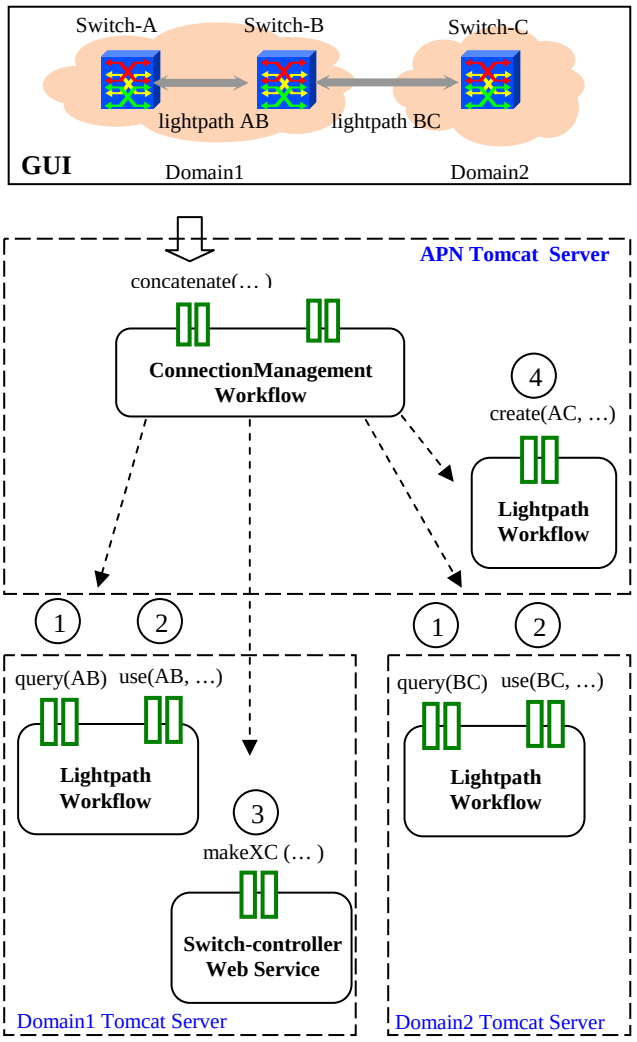

Figure.8 ConnectionManagement Workflow

In Fig. 8 we illustrate what happens behind the scene when a user requests to concatenate lightpaths AB and BC, which are acquired from two different physical networks. In Fig. 8 a dashed arrow represents a BPEL partner link of the ConnectionManagement Workflow. All partner links are dynamic.

The ConnectionManagement Workflow works in four stages, as indicated by numbered circles in Fig. 8. The first stage is a validation stage where all lightpaths and interfaces are queried to validate their operational status, ownership, connectivity, bandwidth, and time-to-live, etc., to see if it is feasible to link them together for the specified user and time period. In the second stage, all participating lightpaths and interfaces are reserved. In the third stage, cross-connections are made sequentially on the switches involved. Finally in stage 4 a super lightpath representing the result of the concatenation gets created in the APN's BPEL engine.

The ConnectionManagement Workflow has a fault handler that catches any exceptions thrown back from switch-controllers during stage 3. When an error occurs, all partial work will be cleaned up: all resources will be un-reserved, and all cross-connections will be undone.

The PartitionManagement Workflow essentially coordinates three activities: terminating lightpaths to be partitioned or bonded, instructing both switch-controller web services to update their partition tables, and finally creating new lightpath partitions or new bonded lightpath.

4.2 Custom Workflows

The case for having custom workflows is as follows. A Grid user may require different usage scenarios for his APN. Each scenario consists of one or more connections within the APN. There will be at most one scenario active at a time, and different scenarios could overlap on lightpath and interface resources. A Grid user is not necessarily an expert in networking or web services. Instead of requiring him to understand the technical details behind reconfiguring an APN, a custom workflow allows a user to switch quickly between predefined scenarios, by simply making reference to a scenario identifier.

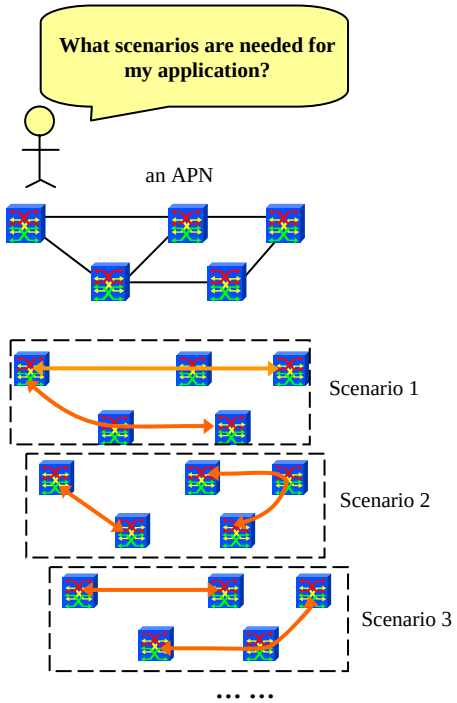

Figure 9 Defining custom usage scenarios of an APN

To create a custom workflow, the user himself or an admin user shall first define different scenarios from the UCLP system GUI, as shown in Fig. 9. UCLP serves as an authoring tool in that defined scenarios will not be provisioned right away. Instead the UCLP system would generate the BPEL file of the custom workflow, as shown in Fig. 10. The XSLT (XSL Transformation) engine transforms a template BPEL file into a valid and complete custom workflow BPEL file, based on the user's input during the authoring stage. The UCLP system would bundle the custom workflow BPEL file with appropriate WSDL files and deployment description files, so that the user could deploy the custom workflow in a BPEL engine of his choice.

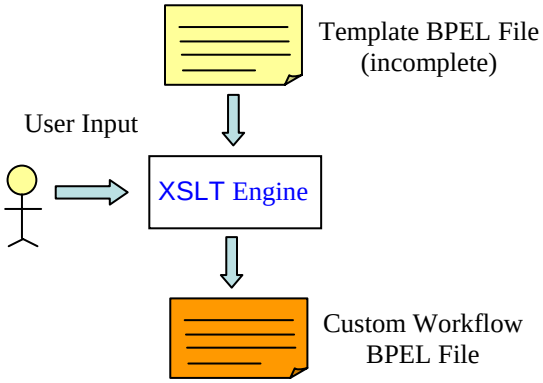

Figure 10 Auto-generation of Custom Workflow BPEL File

We depict the BPEL logics of a custom workflow in Fig. 11, where a white rectangle represents a custom workflow operation, and an arrow represents a web service invocation. Essentially a custom workflow abstracts a re-configurable VPN as one single web service. A Grid user could deploy a custom workflow outside the UCLP system, and orchestrate it with non-network web services into a higher-level workflow, where applications and the network are seamlessly integrated.

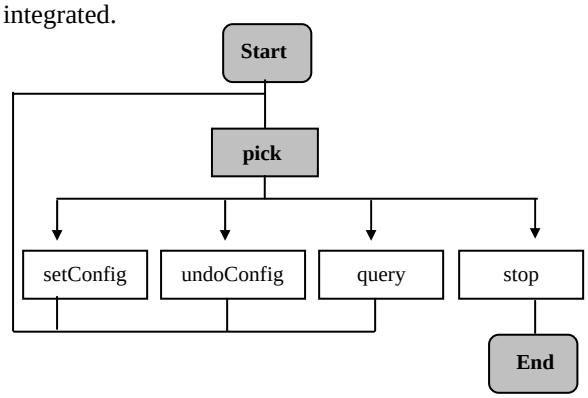

Figure 11 BPEL logics of a custom workflow

As a proof-of-concept, the custom APN concept has been first demonstrated under the Participatory Design Studio project (PDS a.k.a. Eucalyptus [18]) at the Carleton Immersive Media Studio, in December 2006. The PDS project aims at allowing architects and industrial designers at multiple locations to collaborate in real-time by sharing computational resources, geometry datasets, and multimedia content across Canada's CANET*4 research and educational network.

During the demonstration, two custom APN scenarios are defined based on the needs of the architects in Ottawa and Montreal. In the first scenario, two 1.2 Gbps lightpaths are to be set up between the optical cross-connects in Ottawa and Montreal, such that two video sessions can be streamed from Montreal to Ottawa, one Standard Definition (SD) session running at 270 Mbps using Pleora's equipment [19], one High Definition (HD) session running at 880 Mbps using the UltraGrid technology [20]. The video streams will allow architects to view architectural blueprints at the other site in medium or fine details. In the second scenario, a loopback connection needs to be created between a pair of interfaces on the Ottawa cross-connect. The loopback connection traverses many optical cross-connects inside the CANET*4 network, to the extent that the transmission delay is equivalent to a connection spanning thousands of miles. The idea is to simulate the rendering and visualization of Three-Dimensional (3D) architectural models in a long-haul environment, using IBM's Deep Computing Visualization (DCV) middleware [21].

The Pleora and UltraGrid video equipments, the IBM DCV middleware are all controlled through web service interfaces. Using the PDS software, architects were able to orchestrate these web services with the custom APN web service created from UCLP, and switch between the two scenarios promptly. From the architects' perspectives, the network and various equipment and applications are seamlessly integrated.

5. Conclusions

In this paper, we presented a workflow-oriented UCLP system for creating and controlling APNs. In UCLP physical networks are partitioned into virtual network resources, which are then assigned to Grid applications so that they can build their APNs. An APN can either be controlled by human Grid users using utility workflows, or by non-network Grid services and applications via custom workflow interfaces. The ability to create custom APNs on-the-fly is a key feature of the proposed UCLP system.

The proposed UCLP system has been implemented and deployed on the CANET*4 network. We are collaborating with the Grid and e-science community on integrating the custom APNs with other web service enabled applications. We are also working on adding enhanced authentication and authorization to the UCLP system.

5. Acknowledgements

The authors would like to thank the following individuals for their contributions to the UCLPv2 research project: Eduard Grasa, Joaquim Recio and Albert López from the i2CAT Foundation, Mathieu Lemay from Inocybe Technologies Inc., Jing Wu and Bobby Ho from CRC, Mintao Si and Qi Wang from the University of Ottawa, Ramiro Liscano from the University of Ontario, and Hervé Guy from CANARIE.

6. References

[1] V. Sander, W. Allcock, P. CongDuc and et. al., “Networking Issues for Grid Infrastructure”, Global Grid Forum, 2004 http://www.ggf.org/documents/GFD.37.pdf

[2] M. Z. Hasan, W. Clark, M. Morrow, GGF Grid High-Performance Networking Research Group, "Network Service Interfaces to Grid", May 2004 http://www.cs.toronto.edu/~zmhasan/ggf11-nsi.pdf

[3] UCLP web site: http://uclp.ca

[4] R. Boutaba, W. Golab, Y. Iraqi and B. St. Arnaud, "Lightpaths on Demand: A Web-Services-Based Management System", IEEE Communications Magazine, Special Issue on XML-based Network Management, July 2004

[5] R. Boutaba, W. Golab, Y. Iraqi, T. Li and B. St. Arnaud, "Grid-Controlled Lightpaths for High Performance Grid Applications", Journal of Grid Computing, Special Issue on High Performance Networking, Vol. 1, No. 4, pp. 387-394, 2003

[6] B. St. Arnaud, A. Bjerring, O. Cherkaoui, R. Boutaba, M. Potts and W. Hong, "Web Services Architecture for User Control and Management of Optical Internet Networks", Proceedings of the IEEE, Vol. 92 No. 9, pp. 1490-1500, September 2004

[7] Publications and links related to network virtualization at Washington University in St. Louis: http://www.arl.wustl.edu/netv/contributions.html

[8] IETF Layer 1 Virtual Private Networks work group: http://tools.ietf.org/wg/l1vpn/

[9] B. St. Arnaud, "UCLP Roadmap Document", http://www.canarie.ca/canet4/uclp/UCLP_Roadmap.doc

[10] Business Process Execution Language for Web Services version 1.1: http://ifr.sap.com/bpel4ws/

[11] ActiveBPEL http://www.active-endpoints.com/active-bpel-engine-overview.htm

[12] T. Lehman, J. Sobieski, and B. Jabbari, “DRAGON: A Framework for Service Provisioning in Heterogeneous Grid Networks”, IEEE Communications Magazine, Vol. 44, Issue 3, pp. 84-90, March 2006

[13] X. Zheng, M. Veeraraghavan, N. S. V. Rao, Q. Wu, and M. Zhu, “CHEETAH: Circuit-switched High-speed End-to-End Transport ArcHitecture Testbed”, IEEE Communications Magazine, Vol. 43, Issue 8, pp. 11-17, Aug. 2005

[14] The OSCARS Project: http://www.es.net/oscars/

[15] L. Gommans, F. Dijkstra, C. de Laat, and et. al., “Application Drive Secure Lightpath Creation Across Heterogeneous Domains”, IEEE Communications Magazine, Vol. 44, Issue 3, pp. 100-106, March 2006

[16] Juniper JUNOS routing instances overview: http://www.juniper.net/techpubs/software/junos/junos74/swconfig74-routing/html/instance-overview.html

[17] Apache Axis: http://ws.apache.org/axis/

[18] Eucalyptus project home at the CIMS: http://www.cims.carleton.ca/60.html

[19] Pleora website: http://www.pleora.com/

[20] UltraGrid website: http://ultragrid.east.isi.edu/

[21] IBM Deep Computing Visualization: http://www-03.ibm.com/servers/deepcomputing/visualization/