Raytheon
BBN Technologies

UCDAVIS
UNIVERSITY OF CALIFORNIA

Final Scientific/Technical Report

DOE Award Number:	DE-SC0010619/DE-SC0010727
Recipients:	Raytheon BBN Technologies/UC Davis
Project Title:	Exascale Virtualized and Programmable Distributed Cyber Resource Control
Principal Investigators:	Gregory S. Lauer (Lead PI)
	S. J. Ben Yoo (Co-PI, UC Davis)
Date of Report:	30-Sept-2016
Period of Performance:	01-Sep-2013 through 30-Aug-2016

Prepared For:

US Department of Energy

1000 Independence Avenue, SW

Washington, DC 20585

Attn: Teresa Beachley; Thomas D. Ndousse-Fetter

BBN Identifier: EXSCL-FSTR-003-01

This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

Acknowledgment: This material is based upon work supported by the Department of Energy under Award Number(s) DE-SC0010619/DE-SC0010717.

1 Executive Summary

Extreme-science drives the need for distributed exascale processing and communications that are carefully, yet flexibly, managed. Exponential growth of data for scientific simulations, experimental data, collaborative data analyses, remote visualization and GRID computing requirements of scientists in fields as diverse as high energy physics, climate change, genomics, fusion, synchrotron radiation, material science, medicine, and other scientific disciplines cannot be accommodated by simply applying existing transport protocols to faster pipes. Further, scientific challenges today demand diverse research teams, heightening the need for and increasing the complexity of collaboration.

To address these issues within the network layer and physical layer, we have performed a number of research activities surrounding effective allocation and management of **elastic optical network (EON)** resources, particularly focusing on FlexGrid transponders. FlexGrid transponders support the opportunity to build Layer-1 connections at a wide range of bandwidths and to reconfigure them rapidly. The new flexibility supports complex new ways of using the physical layer that must be carefully managed and hidden from the scientist end-users. FlexGrid networks utilize flexible (or elastic) spectral bandwidths for each data link without using fixed wavelength grids. The flexibility in spectrum allocation brings many appealing features to network operations. Current networks are designed for the worst case impairments in transmission performance and the assigned spectrum is over-provisioned. In contrast, the FlexGrid networks can operate with the highest spectral efficiency and minimum bandwidth for the given traffic demand while meeting the minimum quality of transmission (QoT) requirement.

Two primary focuses of our research are: (1) resource and spectrum allocation (RSA) for IP traffic over EONs, and (2) RSA for cross-domain optical networks. Previous work concentrates primarily on large file transfers within a single domain. Adding support for IP traffic changes the nature of the RSA problem: instead of choosing to accept or deny each request for network support, IP traffic is inherently elastic and thus lends itself to a bandwidth maximization formulation. We developed a number of algorithms that could be easily deployed within existing and new FlexGrid networks, leading to networks that better support scientific collaboration. Cross-domain RSA research is essential to support large-scale FlexGrid networks, since configuration information is generally not shared or coordinated across domains. The results presented here are in their early stages. They are technically feasible and practical, but still require coordination among organizations and equipment owners and a higher-layer framework for managing network requests.

2 Accomplishments vs. Project Goals and Objectives

This project proposed to investigate and develop an architecture and distributed control systems for large-scale scientific collaboration. In particular, we proposed to develop architecture and algorithms with the following objectives:

1. **Horizontal Integration Objective:** Multiple domains can collaborate
2. **Vertical Integration Objective:** Enable efficient cross-layer management of emerging terabit-scale transport technologies (e.g., FlexGrid)
3. **Virtualization Objective:** Allow multiple users to share infrastructure dynamically and securely without having to worry about the underlying technological details
4. **Programmability Objective:** Allow new services to be quickly deployed and used.
5. **Software Development Objectives:** Prototype and simulate algorithms as they are designed.

Based on early conversations with the DoE program manager, Dr. Thomas Ndousse-Fetter, we narrowed our focus to the network layer, and focused particularly on the challenges and benefits associated with using FlexGrid optical technology. Within the revised scope, we addressed the objectives through a variety of activities.

Horizontal Integration

- In year 1, we designed a completed SDN/OpenFlow controller to support multi-layer multi-domain optical networks.
- In year 1, we proposed a simplified distributed GMPLS signaling mechanism, which outperforms the state-of-the-art PCE/GMPLS-based solution.
- In year 2, we proposed and studied a distributed control scheme utilizing spectral fragmentation-aware RMSA and flexible active/passive reservation mechanism for dynamic multi-domain software-defined EONs.
- In year 2, we studied the impact of 3R and 4R (with modulation format change) regeneration in EONs, particularly valuable in cross-domain optical networks.
- In year 3, we extended a broker on top of opaquely-managed domains to perform per-domain spectrum defragmentation when no feasible transparent end-to-end lightpath can be found for a multi-domain connectivity request.

Vertical Integration

- In year 1, we collaborated with NEC to extend the Distributed Resource Controller to support optical transport and to develop generic graph algorithms that are layer-agnostic and which work cross-layer.
- In year 1, we designed a completed SDN/OpenFlow controller to support multi-layer multi-domain optical networks.
- In year 2, we experimentally demonstrated dynamic OpenFlow (OF) Based lightpath restoration and defragmentation in elastic optical networks (EONs).

- In year 2, we developed algorithms for supporting internet protocol (IP) traffic over EONs, where the routing of the IP traffic is *not* under the control of the EON Software Defined Network (SDN) controller.
- In year 3, we developed RSA algorithms for maximizing fairness in IP-over-optical networks.
- In year 3, we proposed an architecture for sharing geographically distributed computational facilities among several scientific experiments. A cross stratum heterogeneous Broker orchestrates resource reservation in Data Centers, High Performance Computation facilities, and networks belonging to different operators.

Virtualization

- In year 3, we proposed an architecture for sharing geographically distributed computational facilities among several scientific experiments. A cross stratum heterogeneous Broker orchestrates resource reservation in Data Centers, High Performance Computation facilities, and networks belonging to different operators.

Programmability

- In year 1, we designed several defragmentation and fragmentation-aware algorithms to reduce the network blocking probability and spectrum fragmentation in EONs.
- In year 1, we designed impairment-aware path selection algorithms.
- In year 2, we developed algorithms for reallocating resources to advanced reservations that reduces blocking probability.
- In year 3, we investigated moving computation to optimally placed computation points in the network in order to increase performance for the end-user.
- In year 3, we extended a broker on top of opaquely-managed domains to perform per-domain spectrum defragmentation when no feasible transparent end-to-end lightpath can be found for a multi-domain connectivity request.

Software Development

- Throughout the program, we implemented the majority of our algorithms and ran simulations over real and/or randomly generated network topologies.
- Cross-domain experiments were carried out in a distributed field trial set-up connecting premises in three continents.

3 Project Activities

3.1 BBN

3.1.1 Distributed Resource Controller Extensions

During the first 6 months of the project, we collaborated with NEC to extend the Distributed Resource Controller to support optical transport and to develop generic graph algorithms that are layer-agnostic and which work cross-layer. The initial development of the DRC was funded under DoE Award ED-SC0007340 (A framework for supporting survivability, network planning and cross-layer optimization in future multi-domain Terabit networks). Table 1 summarizes this activity.

The DRC was developed to provide control plane abstractions and mechanisms to open up SDN controller to SDN controller communication in the same way that OpenFlow opened up controller to switch communication. The DRC organizes SDN control via coordinated subgraph shadowing. Graphs are used to represent resources and to describe SDN activity. Subgraphs are used to share a subset of a network's resources between SDN controllers. Shadowing provides a means to dynamically update these shared subgraphs.

In particular, we focused on the following activities:

1. Development and implementation of schemas for DWDM optical networks.
2. Development and implementation of generic graph algorithms that can be applied to various different graph abstractions and across multiple abstractions.
3. Development and implementation of a batch transaction mode that simplifies specifying conditional actions (e.g., rollback on failure).

Table 1: Programmatic summary for Distributed Resource Controller extensions

Original Hypotheses	Optical network and multi-layer networks can be managed more reliably with Distributed Resource Controller support.
Approach Used	Extend DRC to support optical networks, add layer-agnostic graph algorithms, and support batch transactions.
Problems Encountered	None
Departure from Methodology	None
Impact on Project Results	Based on discussions with Dr. Ndousse-Fetter, we changed the focus of BBN's effort from a broad attack on management of network, compute and storage resources to a deeper one that focuses on the network layer and in particular to the challenges and benefits associated with using FlexGrid optical technology. Thus, this activity was independent from the rest of the project.

Optical Transport Extension

The initial DRC design assumed the use of OpenFlow packet switches and had developed abstractions (schemas) for representing: traffic classes and forwarding policies; VPNs; Ethernets; and MPLS-TP. Under this task we added additional abstractions and schemas for:

- Optical Channels. This tracks committed and available lambdas as a function of time.
- Optical Data Unit. This tracks committed and available ODU time slots and hierarchical time slots as a function of time.
- Tunnel. This tracks working and backup paths as a function of time.
- Path. This provides a time annotated resource allocation on specific links that make up a connection between endpoints.

These schemas allow the DRC algorithms to reason about which resources are available when setting up VPNs with bandwidth requirements.

Generic Graph Algorithms

One key goal for the DRC was that of providing topology abstractions that would allow the re-use of graph algorithms at multiple different levels and which could be used to find paths that transit multiple levels of abstractions (e.g., packet switches and optical switches). Under this task we developed the following generic graph algorithms:

- Yen's algorithm. This algorithm finds the K-shortest paths through a network. The K shortest paths can be used as input to other algorithms – e.g., ones that attempt to minimize the maximum load on a link, to find paths that can be examined for contiguous subcarriers, etc.
- Max Flow. This algorithm finds link and/or node disjoint paths between a source and destination. This is useful for finding paths for dedicated or shared protection.
- Dijkstra. This algorithm finds the minimum cost path from a source to a destination. The cost can be specified arbitrarily through the use of anonymous functions that specify at run-time how the cost is to be computed.

Batch Transactions

The DRC provides "triggers" that execute actions when changes are made to the graph database. Sometimes triggered actions need to be combined into a "batch transaction" so they can be executed as a group. For example, the setup of an MPLS tunnel consists of many separate actions on the switches involved. Combining these actions allows for simple specification of pre-conditions and post-conditions. For example, a pre-condition could specify that a user's traffic should not be switched to a new path until the whole path has been provisioned. Post-conditions specify actions to be taken if all the batched actions are successful or not. For example, a success post-condition would tear down the old LSP after the new one is configured; a failure post-condition would roll-back the partially set-up LSP.

Batch transactions were integrated into the subgraph shadowing mechanism, so that these actions can occur across multiple domains and across shared resources.

3.1.2 Advanced Reservations

Requests for optical paths can be *instantaneous* or *advanced*. Both types of requests require the network to either accept the request or block (deny) it immediately. However, instantaneous requests require that the network immediately provision accepted requests while advanced requests specify a future time at which an accepted request should be provisioned.

Advanced requests, which we considered in year 2 (summarized in Table 2), thus provide extra degrees of freedom that can be exploited to reduce the blocking probability:

Table 2: Programmatic summary for advanced reservations

Original Hypotheses	Allowing advanced reservations decreases blocking probability.
Approach Used	Design and implement algorithms for tentative assignments of advanced reservations. Allow changes where this would improve the blocking probability.
Problems Encountered	None
Departure from Methodology	None
Impact on Project Results	This activity resulted in improved RSA algorithms when advanced reservations are allowed.

- Requests can be accepted that start in the future even if no resources are available now, as long as resources are available during the interval running from the specified start time to the specified end time (Figure 1 upper).
- While our algorithm assigns resources to accepted requests, these assignments are viewed as tentative and can be revisited if changes to them would allow a new request to be accepted (Figure 1 lower).

The algorithms we developed support FlexGrid optical paths. The basic approach is as follows:

- When a new request arrives, find all overlapping requests and remove the slots allocated to them from the FlexGrid graph
- Attempt to allocate slots to the new request using one of the RSA algorithms

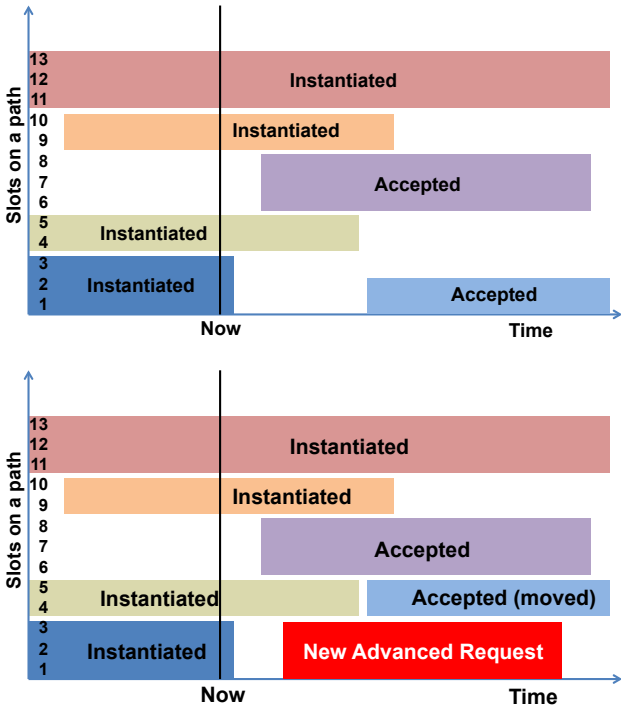

Figure 1. New requests that would have been blocked can be accepted by reassigning resources to requests that have been accepted but which have not yet been instantiated.

- If success, then done; otherwise remove M pending requests from the graph and sort them along with the new request
- Loop through the request list attempting to allocate FlexGrid lightpaths to them
- If success, then accept new request; otherwise deny new request

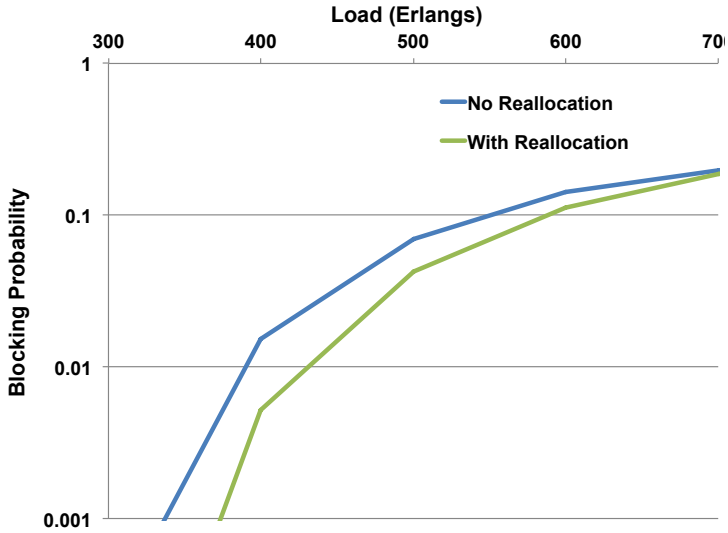

Figure 2. Resource reallocation can decrease blocking probability.

We investigated a number of alternative approaches for identifying a subset of the pending requests including highest weight, earliest start, longest duration, etc. A typical result is illustrated in Figure 2 which compares an approach which attempts to reallocate resources associated with pending requests vs. a baseline algorithm which simply allocates resources as requests arrive. Note that at low loads, blocking probability is reduced by over 50%.

3.1.3 IP and File Transfers

It is unrealistic to assume that all switches and routers are under the control of a single SDN controller – even in a single domain that is controlled by a single administrative authority. For example, there is likely to be significant legacy equipment in the network that cannot be upgraded. In year 2, we investigated a scenario in which the IP routers are *not* controlled by an SDN controller but instead execute a standard routing protocol (e.g., open shortest path first (OSPF)). The goal of the EON SDN controller in this scenario is to build a minimum cost optical network that connects the IP routers and allows them to carry the IP load. This activity is summarized in Table 3.

Table 3: Programmatic summary for IP and file transfers

Original Hypotheses	A hose model is a good representation of IP traffic from the point of view of edge ISPs, and a combination of mixed integer programming techniques and established RSA algorithms can be used to choose lightpaths.
Approach Used	Design and implement algorithms, and test over randomly generated networks.
Problems Encountered	Without knowing the exact routing algorithm to be used at the IP layer (for instance, which shortest path would be chosen from among equal length paths? Is multi-path routing used?), our solution needed to either over-provision or make guesses.
Departure from Methodology	None
Impact on Project Results	The algorithms developed under this activity demonstrated how lightpaths can be generated for one particular model of IP traffic while obeying S-BVT constraints.

There are two models we have identified for characterizing the IP load: the 'pipe' and the 'hose' models. The 'pipe' model specifies the traffic load between each pair of IP routers in the network, while the 'hose' model only specifies the maximum ingress and egress rates for each router. The pipe model is more common in the literature but is much more difficult to implement (one must estimate N^2 flow variables for N routers vs. N flow variables for the hose model). The hose model is easier to characterize and it is possible that the ingress/egress rates can be determined by the router interface speeds or from SLAs. While the hose model leads to a simpler traffic estimation problem, the technical problem of determining a minimum cost optical network is more complex as it must be capable of handling any traffic pattern that is consistent with the hose model constraints.

We developed algorithms for optimizing the optical network when the hose model parameters are known. Our initial algorithm accounts for constraints on the number of lightpaths that can originate/terminate on a Sliceable Bandwidth Variable Transponder (S-BVT) as well as the standard EON continuity and contiguity constraints. It minimizes the optical bandwidth required to support the IP traffic subject to these constraints. The algorithm operates as follows:

- Construct a full-mesh graph where the cost of a link is the number of optical links required to instantiate a lightpath between the pair of nodes.
- Evaluate a mixed Integer Linear Program to determine the lightpath links to instantiate and the bandwidth required on each link.
- Attempt to instantiate the lightpaths using an RSA algorithm. If all lightpaths can be instantiated, then 'success' and the algorithm terminates.
- If not all lightpaths can be instantiated, then remove some links from the full-mesh graph and resolve. If repeated attempts to resolve fail, then reduce the ingress/egress requirements specified by the hose model.

Figure 3 provides an example of a problem solved using this approach. The S-BVTs are limited to using 2 receive and 2 transmit lightpaths. The hose model requirements are listed for nodes 2-5, 9, 12 and 13; the remaining nodes do not generate or receive IP traffic.

The optimal solution for this problem consists of 2 symmetric bi-directional lightpaths (9-13 and 12-13), one asymmetric lightpath (5-12) and a loop of 4 unidirectional lightpaths (connecting nodes 1, 2, 5, 4 and 3).

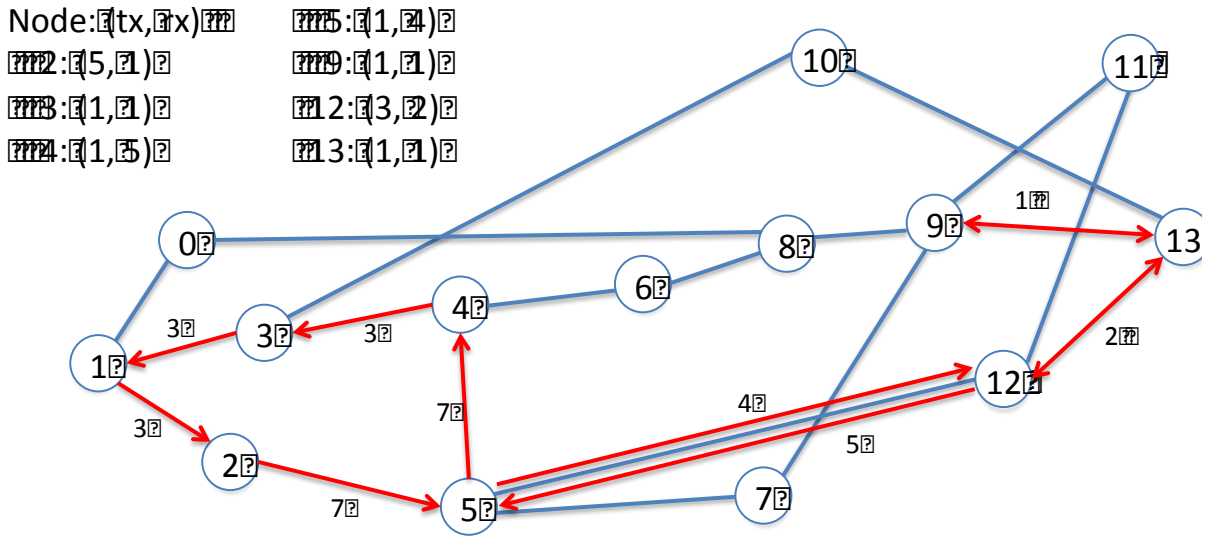

Figure 3. NSF Network with hose constraints. The solution consists of the links marked in red.

3.1.4 Routing and Spectrum Assignment for IP over Optical

The RSA problem has typically been studied in the context of large file transfer applications. In these applications, the primary goal is to accept as many file transfer jobs as possible by minimizing the blocking probability (i.e., the probability that some later job must be rejected). However, FlexGrid networks may be of even more use as the backbone for an IP network, in which small changes to the routing and spectrum assignments can lead to smooth transitions over time to accommodate changing demands on the network. To that end, in year 3 we explored RSA algorithms with the goal of *maximizing fairness* across the network. Specifically, we examined a problem in which we're given a set of pairwise demands $d(s_i, t_i)$ for various source-sink pairs, and our goal is to generate lightpaths assignments that maximize D such that we can simultaneously satisfy $D*d(s_i, t_i)$ for all i. This activity is summarized in Table 4.

Table 4: Programmatic summary - RSA for IP over optical

Original Hypotheses	Maximizing fairness is a reasonable goal for supporting IP traffic, and a heuristic-based algorithm is likely to perform well without sacrificing efficiency.
Approach Used	Design and develop a number of algorithms, and test them all for the same set of randomized input. Compare the quality of the results (D value for scaling demand) as well as the running times.
Problems Encountered	None
Departure from Methodology	None
Impact on Project Results	This activity resulted in a thorough comparison of a number of algorithms for this model of IP traffic. The shortest path algorithm performed best, with maximizing demand divided by path length performing next best.

We implemented and compared the following set of algorithms for maximizing fairness.

First Fit (highest demand first): In decreasing order of demand size, search for the lowest numbered block of slots that can meet the demand. Assign a lightpath along the shortest path with that block of slots available.

First Fit (unsorted demands): As above, except handling the demands in an arbitrary order rather than highest demand first.

Maximize Available Bandwidth: In decreasing order of demand size, pick the lightpath that maximizes the total number of remaining slots from that source to that destination. (Ties broken in favor of shortest path.)

Tightest Packing: Find the tightest fit that can satisfy the required demand.

Maximize Largest Available Block: Find the path that leaves the largest possible unallocated block from that source to that destination.

Maximize Demand/Path Length: Find the shortest available path for each demand pair. Chose the one with the maximum demand/path length. Assign that lightpath and repeat with the remaining demand pairs.

Shortest Path: In decreasing order of demand size, assign each pair to the shortest path available, ties broken by earliest slot first.

Figure 4 shows the results of comparing these algorithms. Over 106 random instances of varying sizes (randomized topology and demands), we executed all 7 algorithms on each instance. The graph shows in how many instances each of the algorithms achieved what fraction of the best result for that instance. There was no correlation between problem size and algorithm performance. Figure 5 and Figure 6 show the comparative running times for the algorithms, compared to the size of the problem being solved. Note that we did not spend time optimizing the algorithms for running time. Based on these results, it appears that using a simple shortest path algorithm (modified based on available slots) is the best heuristic for IP fairness.

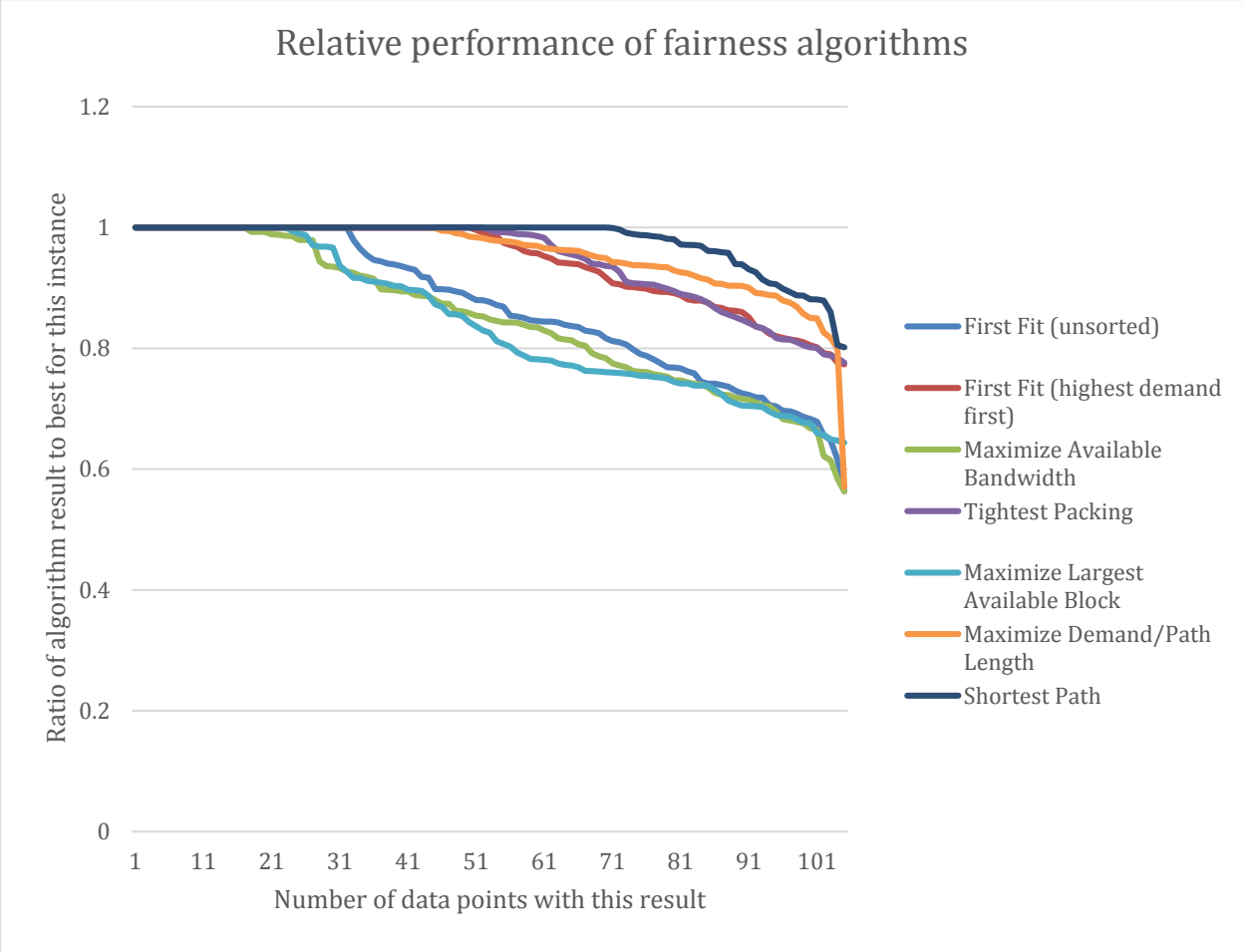

Figure 4: Comparison of heuristic algorithms for assigning lightpaths to optimize fairness for IP traffic.

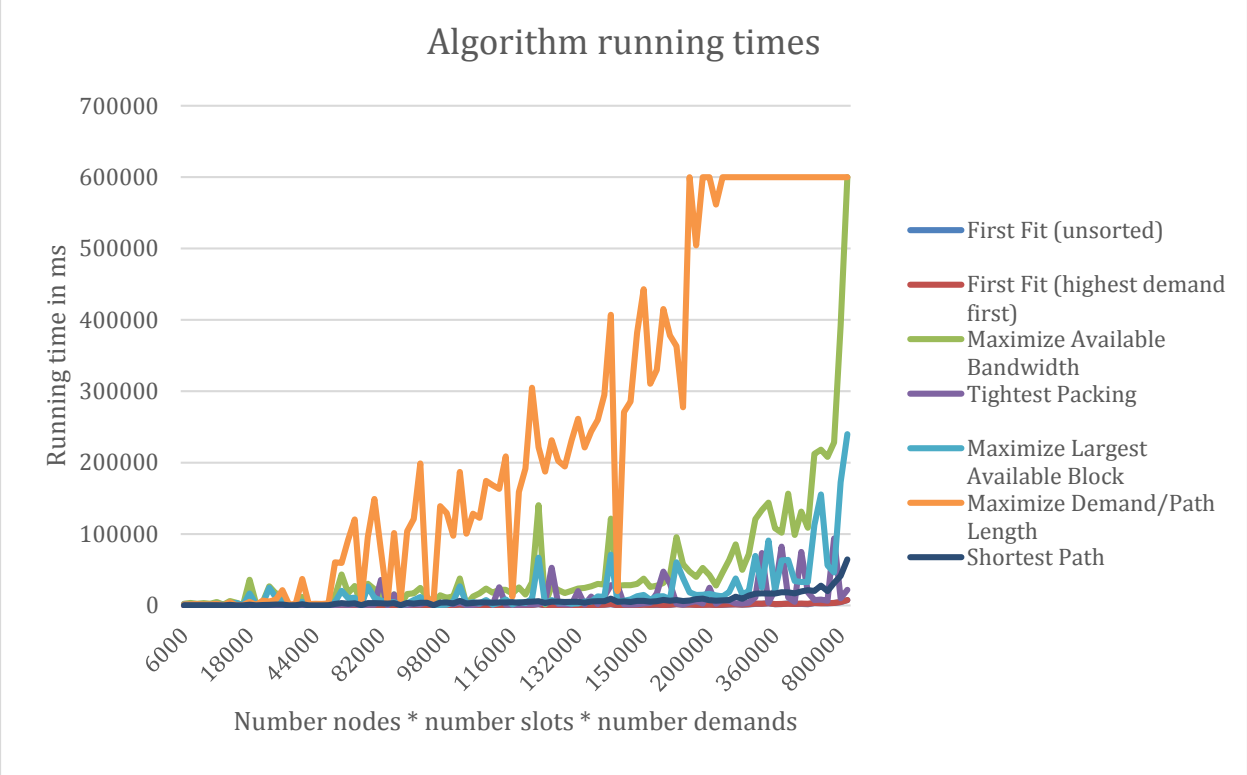

Figure 5: Algorithm running times by problem size for IP fairness algorithms

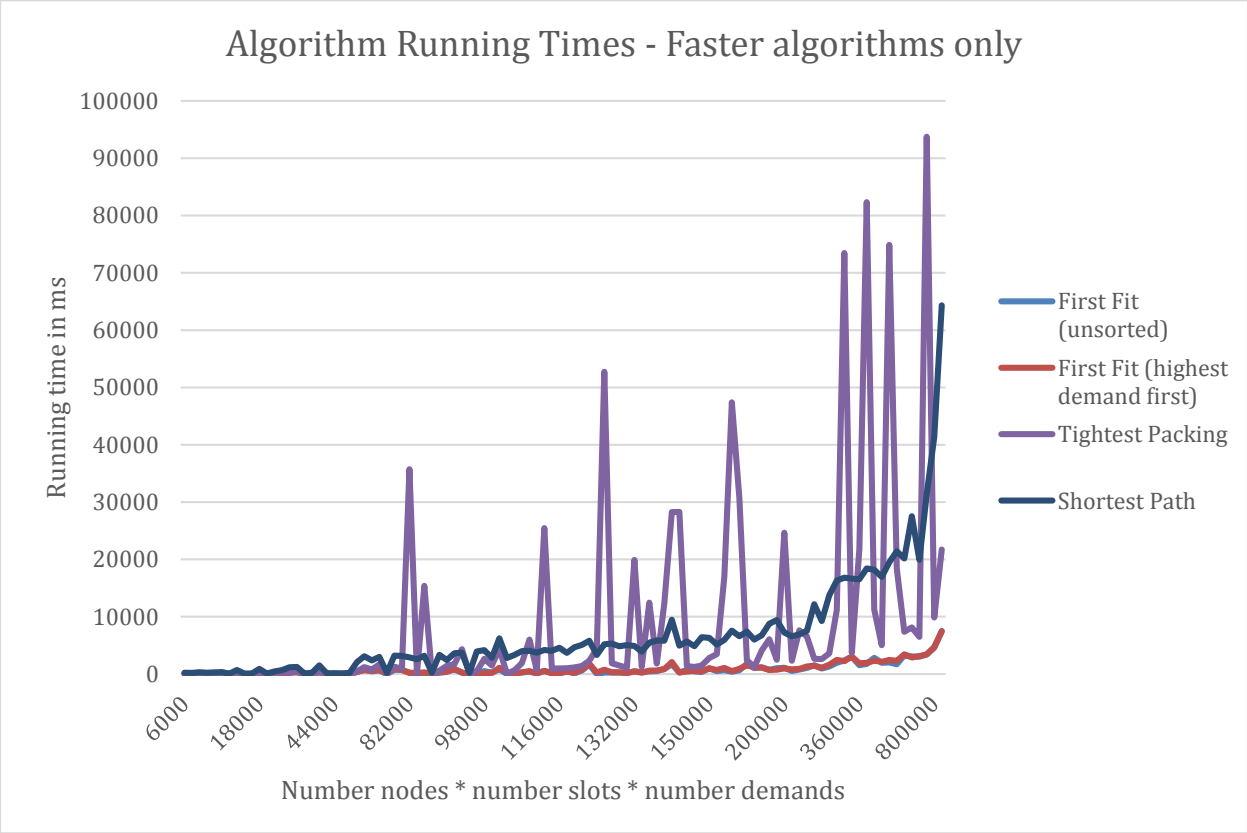

Figure 6: Algorithm running times by problem size for fastest IP fairness algorithms

3.1.5 Exploration of Linear Programs for RSA Problems

We drafted a number of linear and integer program formulations for solving variants of the RSA problem throughout years 2 and 3. This activity is summarized in Table 5. Developing ILP models serves three purposes:

1. Allow the problem to be solved using ILP solver software
2. Allow standard approximation algorithm techniques that use an ILP model as a baseline.
3. Precisely define the problem for the sake of future research.

Table 5: Programmatic summary for exploration of LPs for RSA problems

Original Hypotheses	RSA problems can be written as mixed integer programs.
Approach Used	Develop mixed integer program models, and implement some of them using an open source LP solver.
Problems Encountered	We found that using a non-commercial solver significantly limited the supported problem size.
Departure from Methodology	We originally planned to incorporate a number of real-world S-BVT constraints in our problem formulation. However, due to the limitations from the LP solver, we constrained ourselves to only supporting a subset of the constraints.
Impact on Project Results	The program definitions helped us to better understand the problem, including the complexity of real-world constraints. The Fair-sharing model was a basis for developing one of our heuristic algorithms, and partially mathematically explained the heuristic algorithm results.

We implemented some of these using the python interface to the open source LP solver "lpsolve." In general, these formulations are too large to be solved in a reasonable amount of time by lpsolve, including the programs with no integer constraints. A commercial LP solver would like perform better. Alternatively, these could be used as the basis for a primal-dual algorithm, in which the dual of the linear (or integer) program is solved incrementally, using the dual variable assignments to solve the primal problem. Primal-dual algorithms generally have low running times, are easy to adapt to an online setting or to a distributed solution, and allow for proving approximation ratios against the optimal solution. A primal-dual algorithm investigation lead to the "Maximize Demand/Path Length" algorithm we implemented and discussed above.

The formulations we developed are listed below.

RSA for large file transfers: Given (1) the underlying topology, including a number of real-world constraints on the transponders and links, and (2) a set of file transfer requirements, each including the source node, destination node, required number of slots, start time, and end time. Generate a lightpath for each file transfer (obeying all constraints) to maximize the total number of file transfers accepted into the system. This ILP was documented, implemented, and tested. It reached valid nearly-optimal solutions, but only for very small problem instances.

RSA for IP traffic: Given (1) the underlying topology, including a number of real-world constraints on the transponders and links, and (2) A set of node pairs for which we want to support IP traffic. Find (1) a set of lightpaths that obey all constraints, and (2) the (shortest path in the lightpath graph) route used by each required pair. The goal is to maximize the minimum bandwidth between any required pair. This ILP was documented but not implemented.

Fair-sharing RSA for IP traffic: Given (1) the underlying topology, and (2) a set of node pairs with demand for lightpath with a specified number of slots between each pair. Find lightpaths between all pairs to maximize the minimum fraction of satisfied demand. This ILP would find the optimum solution to the problem discussed in Section 3.1.4. We implemented this and also worked through a number of optimizations to make it work for larger problem instances. We also used some techniques specific to the lpsolve framework that allowed up to find approximate solutions using an LP (as opposed to an ILP).

3.1.6 In-Network Computation

We finished year 3 with a study on locating computation within a network rather than at the edges, as summarized in Table 6. This is an orthogonal exploration to the FlexGrid research, but is also necessary to enable the proposed scalable collaboration infrastructure for extreme-scale science. As long as computation is constrained to the edges of the network, all computation must either be performed at a compute cluster, which requires wasting bandwidth to move data from the edge data sources to the cluster, or at the edge devices themselves, which restricts the available computation power. Thus, existing computation models inherently limit the performance and scalability of the system. In order to perform large data, high speed computation, the computation *must* be done at strategically chosen points in the network.

Table 6: Programmatic summary for in-network computation

Original Hypotheses	Resources will be better utilized if we allow flexibility in determining *where* computation is performed and *how* data traverses the network.
Approach Used	Implement mixed integer-linear programs that find the optimal computation location if (a) computation must be located at only one node and data must travel only one path and (b) computation can be distributed across multiple nodes and data can travel over multiple routes.
Problems Encountered	Open source linear program solvers don't scale well.
Departure from Methodology	Simplified the problem statement to have the data source equal the data destination and avoid specifying tight computation bottlenecks.
Impact on Project Results	This was a promising direction, but more work would be needed to tie this back to the rest of the project.

In this activity, we investigated the potential improvements that are possible by intelligently locating and distributing computation. We used a linear program that maximized the total amount of computation performed subject to bandwidth constraints on links and com-

putation constraints on nodes. Computation could be spread over multiple nodes, reflecting a realistic use case of distributed computation. Input and output data could be routed over many paths, reflecting a network model such as backpressure routing. Figure 7 shows the results of comparing a single-path, single-computation-node model to an in-network distributed computation model allowing all paths to be used. For these experiments, one-fifth of the nodes were designated as data sources and one-fifth as computation nodes.

The open problem remains of how much the computation power could be improved by using a FlexGrid network either instead of backpressure routing or to enable improvements on backpressure routing.

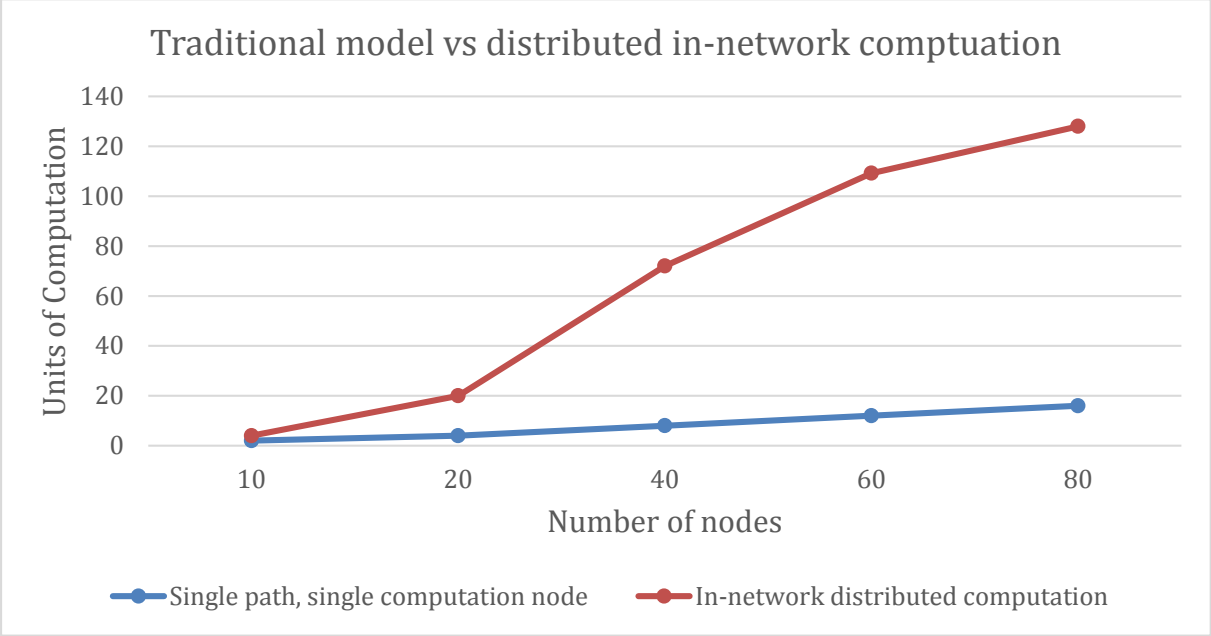

Figure 7: In-network distributed computation enables more effective use of resources.

3.2 UC Davis

3.2.1 Simplified GMPLS control plane

There are two different GMPLS control plane designs so far: pure GMPLS and GMPLS with path computation element (GMPLS/PCE). In pure GMPLS design, each node maintains its own network state information and works for routing, modulation format, and spectrum assignment (RMSA). In contrast, in GMPLS/PCE design, centralized stateless PCE maintains its own network information. However, both GMPLS based EONs present two well-known issues. First, routing protocols require large control plane overhead due to the dissemination of the OSTP-TE protocol. Second, GMPLS distributed signaling cannot avoid reserve collision among requests due to the signaling-latency. In light of this, during year 1 we proposed fully-distributed RMSA and simplified GMPLS signaling with the possibility for reduced collision as an extension to GMPLS. This activity is summarized in Table 7.

Table 7: Programmatic summary for simplified GMPLS control plane

Original Hypotheses	Explore different GMPLS architectures and resource reservation problem.
Approach Used	Implement and simulate a novel RMSA.
Problems Encountered	None
Departure from Methodology	None
Impact on Project Results	Our proposed network achieves significantly better performance in reservation collisions avoidance.

The solution contains three key steps: (1) look for K routes (K is a pre-determined number) from the source node to the destination node using Path message broadcasting; (2) attempt to establish multiple paths, and (3) tentatively reserve slots for later arriving Path messages to avoid over-reservation. Numerical results show that our proposed network, when K is two or more, achieves much better performance in terms of reducing the blocking probability and signaling delay than does the state-of-the-art GMPLS/PCE network. In addition, our proposed network achieves significantly better performance in the case of the possibility of reservation collisions being large, such as a large scale network, due to the proposed tentative reservation.

3.2.2 SDN/OpenFlow controller design

In year 1, we designed a SDN/OpenFlow controller capable of fragmentation-aware RMSA computation and elastic path provisioning. This activity is summarized in Table 8.

Table 8: Programmatic summary of SDN/OpenFlow controller design

Original Hypotheses	Reduce fragmentation by improving the RSMA in SDN optical networks.
Approach Used	Devise and implement a fragmentation aware RMSA
Problems Encountered	None.
Departure from Methodology	None.
Impact on Project Results	Our proposed fragmentation-aware RSMA algorithms, can greatly reduce the blocking probability in EONs.

The designed controller can parse the extended OpenFlow *Packet In* messages from the client layer, and get the flow information including source address, destination addresses and required bandwidth. Then based on the traffic engineering database (TED), the controller performs fragmentation-aware RSMA algorithms to intelligently route the incoming requests to avoid spectrum fragmentation in EONs. The extended *Flow Mod* message carries the RMSA results from the controller, including input and output ports, central frequency, number of spectrum slots, and modulation format for each OpenFlow agent to control the underlying data plane hardware. We proposed two algorithms here, which are referred to as Minimum Path Cut (MPC) and the Minimum Path Cut with Network Resource Optimization (MPC-NRO) algorithm respectively. The evaluation results show that our designed controller, together with our proposed fragmentation-aware RSMA algorithms, can greatly reduce the blocking probability in EONs.

3.2.3 Impairment-aware path selection algorithms

In year 1, we proposed an adaptive quality of transmission (QoT) path selection / restoration schemes combining the methods of lightpath rerouting and modulation-format switching to overcome real-time impairments in EONs (summarized in Table 9).

Table 9: Programmatic summary of impairment-aware path selection algorithms

Original Hypotheses	Improve the QoT by taking into account the physical impairments.
Approach Used	Develop a two phase algorithm impairment aware RSMA.
Problems Encountered	None.
Departure from Methodology	None.
Impact on Project Results	The algorithm achieved more than 50% reduction in blocking probability and 75% decrease in the spectral consumption.

A hybrid objective algorithm is designed to minimize the spectral consumption and the interference that the rerouting imposes on the other parts of the network. Simulation results show the algorithm achieved more than 50% reduction in blocking probability and 75% decrease in the spectral consumption on the bypass links compared with the previous method. We also conducted testbed experiments, which demonstrated successful real-time QoT path provisioning / restoration in the presence of time-varying optical signal-to-noise ratio impairment for two simultaneously impaired 100 and 200 Gb/s quaternary phase-shift keying super-channels. In addition, we also proposed a two-step RSMA algorithm, which firstly generates static planning table with the transmission performances by using simulations, and then performs dynamic resource assignment based on the static planning table to guarantee the QoT.

3.2.4 Network defragmentation algorithms

In year 1, we addressed the spectral defragmentation problem using an auxiliary graph based approach, which transforms the problem into a matter of finding the maximum independent set (MIS) in the constructed auxiliary graph. This activity is summarized in Table 10.

Table 10: Network defragmentation algorithms

Original Hypotheses	Reduce blocking probability by applying spectral defragmentation.
Approach Used	Approach the defragmentation problem with a multi-graph structure.
Problems Encountered	None.
Departure from Methodology	None.
Impact on Project Results	Reduced the blocking probability in EONs.

The enabling technologies and defragmentation-capable node architectures, together with heuristic defragmentation algorithms are proposed and evaluated. Simulation results show that the proposed min-cost defragmentation algorithms can significantly reduce the blocking probability of incoming requests in a spectrally fragmented flexible bandwidth optical network, while substantially minimizing the number of disrupted connections. Moreover, since the defragmentation is complicated from the network operational point of view, it is always beneficial to perform fragmentation-aware and alignment-aware RMSA during provisioning to minimize the need for future defragmentation. In light of this, we proposed two spectral and spatial 2D fragmentation-aware and alignment-aware RMSA to reduce the blocking probability in EONs.

3.2.5 Openflow-Based Lightpath Restoration and Defragmentation in Elastic Optical Networks

In year 2, we investigated OF-based implementations for realizing online defragmentation (DF) in both single- and multi-domain SD-EONs, as summarized in Table 11.

Table 11: Programmatic summary of Openflow-based lightpath restoration and defragmentation in EONs

Original Hypotheses	OF-based implementations for realizing online defragmentation in both single- and multi-domain SD-EONs
Approach Used	OF protocol extensions to support efficient online defragmentation.
Problems Encountered	None.
Departure from Methodology	None.
Impact on Project Results	The proposed defragmentation system performed well and could improve the performance of SD-EONs effectively.

We performed the overall system design and OF protocol extensions to support efficient online DF, and conducted DF experiments with Routing and Spectrum Allocation(RSA) re-configurations in a single-domain SD-EON. Then, we studied how to realize Fragmentation Aware RSA (FA-RSA) in multi-domain SD-EONs with the cooperation of multiple OpenFlow controllers (OF-Cs). We designed and implemented an Interdomain Protocol (IDP) to facilitate FA-RSA in multi-domain SD-EONs, and demonstrated controlling the spectrum fragmentation on inter-domain links with FA-RSA. Our experimental results indicated that the proposed DF systems performed well and could improve the performance of SD-EONs effectively. For more details please refer to paper [3] in the publication list below.

We also investigated OF-based dynamic lightpath restoration in EONs. We proposed an OF-based failure isolation mechanism and a new OF OFPT_ALARM message. Based on these, a two-phase restoration routing, spectrum, and modulation format assignment (RSMA) algorithm is presented. More importantly, the overall feasibility and efficiency of the proposed solutions, including the control framework, the failure isolation mechanism, the restoration algorithm, and protocol extensions are validated and quantitatively evaluated in terms of restoration latencies and restorability on the Global Environment for Network Innovations (GENI) testbed, which includes many GENI racks, regional and national backbone networks around the United States. This allows validating the overall feasibility of the approach and provides valuable insights into its potential for possible deployment in the future. For more details, please refer to paper [4] in the publication list below.

3.2.6 Distributed Control Plane with Spectral Fragmentation-Aware RSMA and Flexible Reservation for Elastic Optical Networking

This year 2 activity is summarized in Table 12.

Table 12: Programmatic summary for EON distributed control plane

Original Hypotheses	Distributed Control Plane with Spectral Fragmentation-Aware RSMA.
Approach Used	Propose an RSMA algorithm that utilizes a distributed breadth first search algorithm to find multiple feasible route candidates.
Problems Encountered	None.
Departure from Methodology	None.
Impact on Project Results	The proposed solution outperforms the conventional GMPLS/PCE scheme.

The proposed RSMA algorithm utilizes a distributed breadth first search algorithm to find multiple feasible route candidates. As a request arrives, domain controllers calculate the shortest route to every neighbor domain, update the 3 information fields and send the updated request message to that neighbor domain (see **Error! Reference source not found.** (top)). The destination domain controller will select K+1 feasible shortest paths based on the total distance field and performs modulation format selection and spectrum slots assignment. On each path, the free spectrum slot band which contains the slot with the minimal spectrum cut factor, f_i, will be selected to assign the lightpath.

Error! Reference source not found. (bottom) summarizes the three cases of passive reservation process. The destination domain controller applies active reservation for the shortest path among (K+1) feasible path candidates found and uses passive reservation for other K paths. The active reservation is similar to RSVP-TE; each domain reserves its necessary intra-domain resources and then it sends backward a RESERVE message. If the intra-domain resource reservation is failed, a PathERR and a RELEASE will be provided backward (to the source domain) and forward (to the destination domain) respectively. Passive reservation process is deployed for other K feasible paths in order of time sequence or total distance. When the passive reservation message (PASS RESERVE) arrives the source domain while the active reservation was failed, the domain controller will send out the ACTIVATE message to activate the passive-reserved path. Otherwise, a RELEASE message is sent out to clear the passive-reserved path. When the ACTIVATE message arrives the destination domain, the destination domain controller will feedback an ACK message to the source domain to inform about the activation success, otherwise the domain with the intra-domain resource activation failure will send a RELEASE message to both directions and transmit a PathERR backward to the source domain.

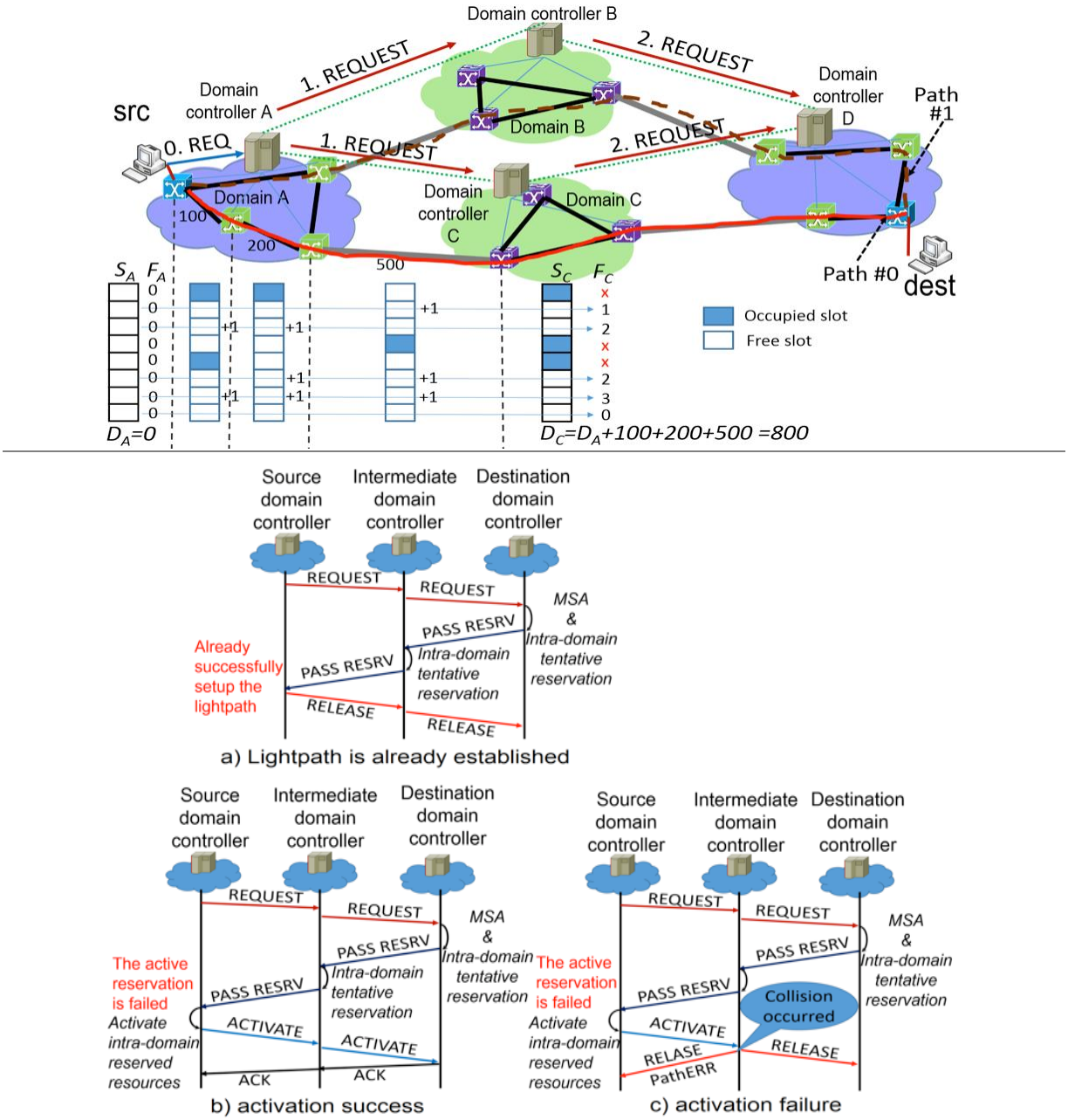

Figure 8: (Top) Distributed fragmentation-aware RMSA. (Bottom) Passive reservation process

Error! Reference source not found. illustrates the blocking probability of the developed control scheme with the applied number of passive reservations per request, *K*, of 1 and 2 when the average connection holding times (MHT) are 100 and 1000 respectively. The results prove that the proposed solution outperforms the conventional Generalized Multi-Protocol Label Switching (GMPLS) / Path Computation Element (PCE) scheme. Our proposed solution significantly reduces the call loss probability; up to 40% (52%) smaller blocking probability can be obtained with *K*=1 (*K*=2) when the relative traffic load is 0.5 and MHT=1000.

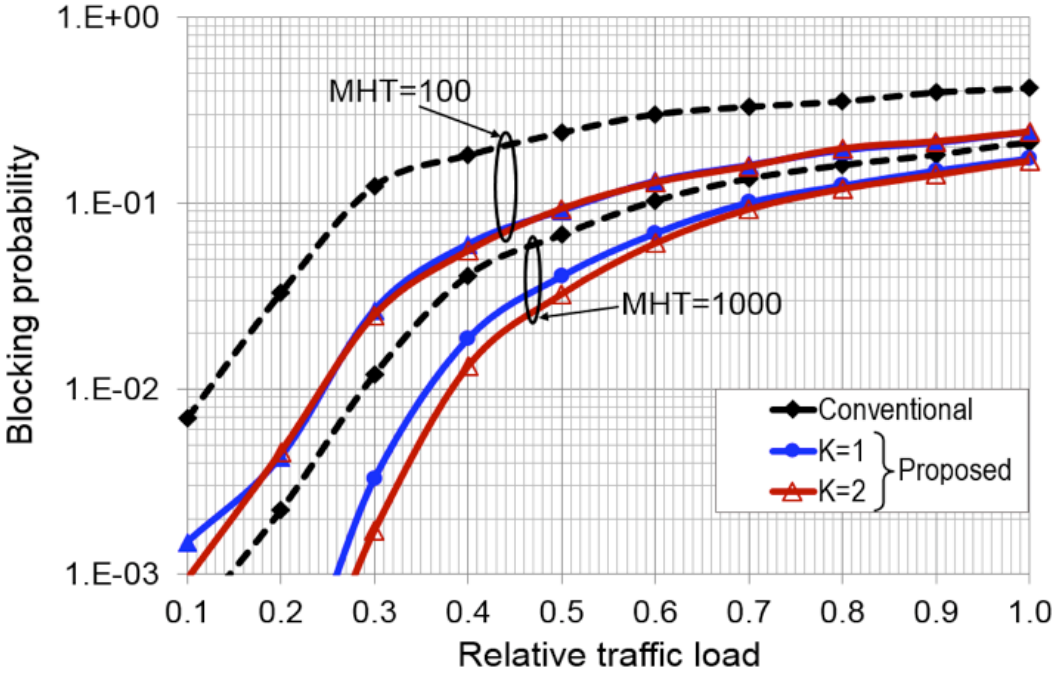

Figure 9: Blocking probability for 24-node US nationwide network

3.2.7 Regeneration Capability in Elastic Optical Networking

In year 2, we studied the effect of 3R and 4R (with modulation format switching) regeneration in EON as function of the number of regenerators per node (summarized in Table 13).

Table 13: Programmatic summary of Regeneration Capability in EON

Original Hypotheses	Studying the effect of 3R and 4R (with modulation format switching) regeneration in EON.
Approach Used	Devise and implement RSMA with 3R and 4R capabilities.
Problems Encountered	None.
Departure from Methodology	None.
Impact on Project Results	Analytic results on the improvements that can be achieved by applying regeneration.

In this study we assume that only a few nodes (border nodes in **Error! Reference source not found.**) have limited regeneration resources, which means that the number of flexible channel slots that they can regenerate at the same time is < *K.*

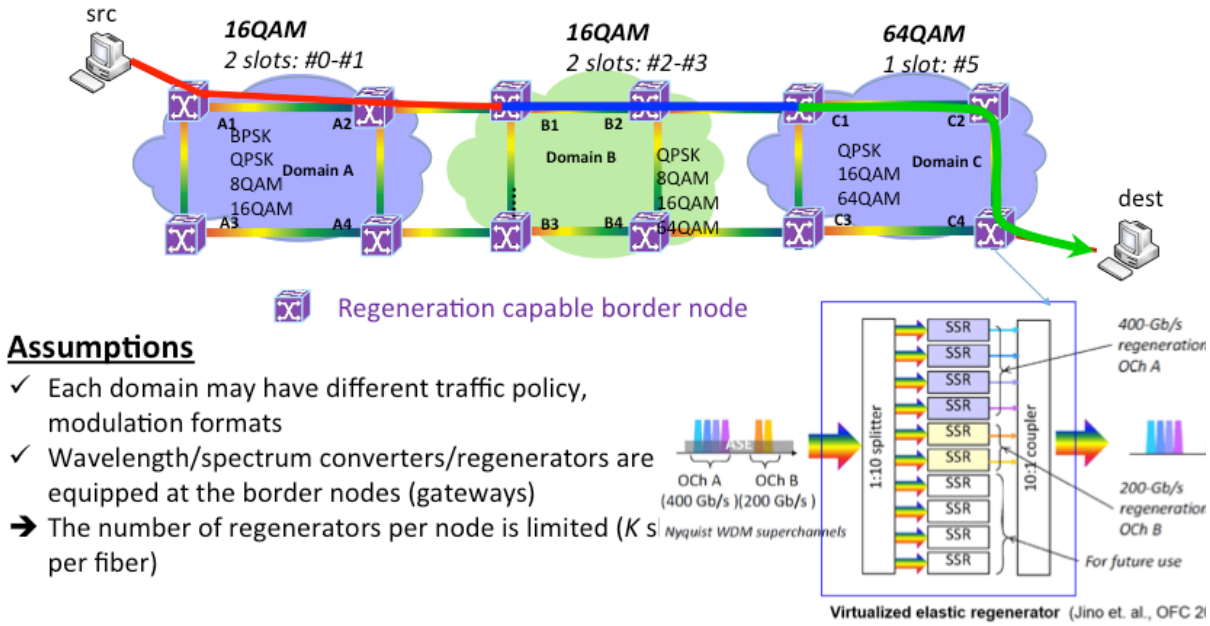

Figure 10: Elastic optical network with regeneration-capable nodes

Error! Reference source not found. shows preliminary results in terms of blocking probability and average spectrum utilization per connection request in case of a typical network file system (NFS) network topology, when considering the modulation formats, spectrum slot capacity and maximum reach indicated in Table 14. For these results we did not consider any limitation in terms of regeneration resources ($K >> 1$).

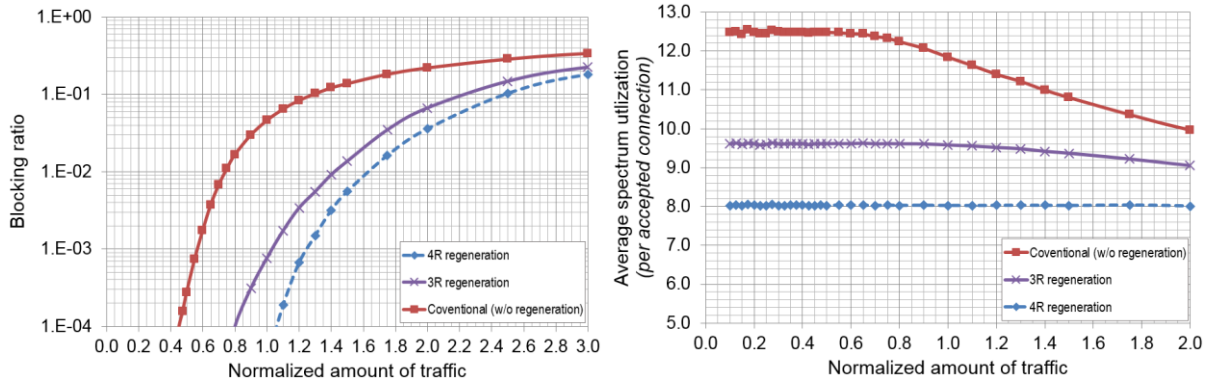

Figure 11: (Left) Blocking probability. (Right) Average spectrum utilization per connection request.

Note that the limited improvement given by the use of modulation format switching in 4R regeneration is related to the fact that most of the internode links are longer than the maximum reach achievable with 16 quadrature amplitude modulation (QAM). Therefore, only few connections can be regenerated while occupying less amount of spectrum. We are currently working on evaluating the case for *K=2, 4, 8, 16* and *32.*

Table 14: Modulation formats and related spectrum slot capacity and maximum reach

Modulation format	Spectrum slot capacity	Distance limit
BPSK	12.5 Gb/s	9600 km
QPSK	25 Gb/s	4800 km
8QAM	37.5 Gb/s	2400 km
16QAM	50 Gb/s	1200 km

3.2.8 Experimental Demonstration of Brokered Orchestration for end-to-end Service Provisioning and Interoperability across Heterogeneous Multi-Operator (Multi-AS) Optical Networks

Flexgrid elastic optical networking (EON) is a promising technique for future metro/core optical networks. To control EONs, Software-defined Networking (SDN) has been widely studied in recent years, in particular when based on the OpenFlow (OF) protocol for its open interface and flexibility in terms of network control and programming. The IETF has been working on a similar approach and recently standardized the Application-Based Network Operations (ABNO) architecture. Previous works on such a software-defined elastic optical networking (SD-EON) focused on single/multi-AS scenarios under the single operator premise However, multi-AS networking architectures are very relevant in real operational scenarios to enhance network scalability and service reach. Therefore, how to support a multi-AS with multiple operators SD-EON is an important topic and needs to be carefully investigated. Note that each operator advertises partial information regarding the topology and connectivity of its AS.

A broker-based SDN solution was proposed in year 3 (Table 15), where a broker is introduced on top of all the SDN controllers to coordinate end-to-end resource management and path provisioning. The centralized broker updates the virtual network topology, manages the resource information of inter-AS links and aggregated (abstracted) intra-AS links, and computes end-to-end routing, modulation formats, and spectrum assignment (RMSA).

Notwithstanding, due to the different dynamicity of each AS, the probability of finding a multi-AS transparent path fulfilling the spectrum continuity constraint might be low. Therefore, per-AS defragmentation can be performed with a global view. In this paper, we propose a mechanism where each AS advertises its internal capabilities, e.g. their ability to implement spectrum defragmentation or any other in-operation planning operation. A planning tool connected to the broker is used to decide the optimal set of operations to provision end-to-end paths.

Table 15: Programmatic summary for brokered multi-operator networks

Original Hypotheses	Provide end-to-end services between black domains.
Approach Used	Introduce a Broker orchestrator and devise a workflow.
Problems Encountered	None.
Departure from Methodology	None.
Impact on Project Results	Experimentally validated a new workflow managed by the broker to provision a multi-AS optical path.

3.2.8.1 Broker-based Multi-Operator Architecture

Let us assume a multi-operator multi-AS flexgrid optical network, where each AS is managed by an SDN/OF controller or an ABNO-based architecture. On top of the ASs, a broker coordinates end-to-end multi-AS provisioning (Figure 12).

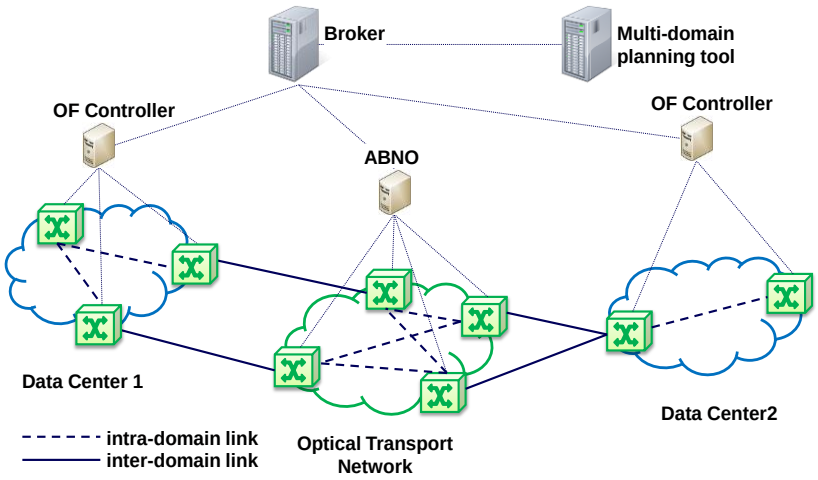

Figure 12: Multi-AS architecture

Each AS advertises an abstracted intra-AS link information to the broker that depends on both, internal AS policies and the specific agreement with the broker. The broker has a global view of the virtualized network topology, including full information of the inter-AS links and abstracted intra-AS link status gathered from each AS.

In addition, an AS may agree to expose further features to the broker. For example, some ASs may have deployed specific hardware (e.g., wavelength converters/regenerators) and/or implemented optimization algorithms (e.g., spectrum defragmentation algorithms), named as *capabilities*.

To model the underlying data plane, let us assume a graph G(N, E), where N is the set of optical nodes and E is the set of optical links connecting two nodes. Graph G is structured as a set of ASs D. Every AS d consists of three differentiated subset of nodes:

- N_e: subset of edge nodes, end-points of demands;
- N_t: subset of internal AS nodes;
- N_i: subset of border AS nodes. Then, $N = N_e \cup N_t \cup N_i$ with $N_e \cap N_i = \emptyset$.
- Let S be the set of available frequency slices in each optical link.

Regarding the links, two subsets are considered:

- E_i: subset of inter-AS links, connecting two nodes in N_i belonging to two different ASs;
- E_n: subset of abstracted intra-AS links. Each $e \in E_n$ abstracts connectivity between either a node in N_e and another node in N_i belonging to the request's end ASs, or between two nodes in N_i belonging to transit ASs.

Each link e is represented by a tuple $<a_e, z_e, S_e, c_e>$, where $a_e, z_e \in N_e \cup N_i$ are the end nodes, S_e is the subset of available frequency slices, and c_e is the cost.

Since both, broker and the planning tool will be requested to perform complex computations, each AS is assumed to advertise sets N_i and E_i at start time, and update the set S for each link in E_i to follow updates, independently from path computation requests. In addition, each AS advertises its capabilities (e.g., spectrum defragmentation) (Figure 13a). When a computation is requested, the broker collects intra-AS data (E_n) (Figure 13b), which are advertised to the planning tool in case that in-operation planning is needed (Figure 13c).

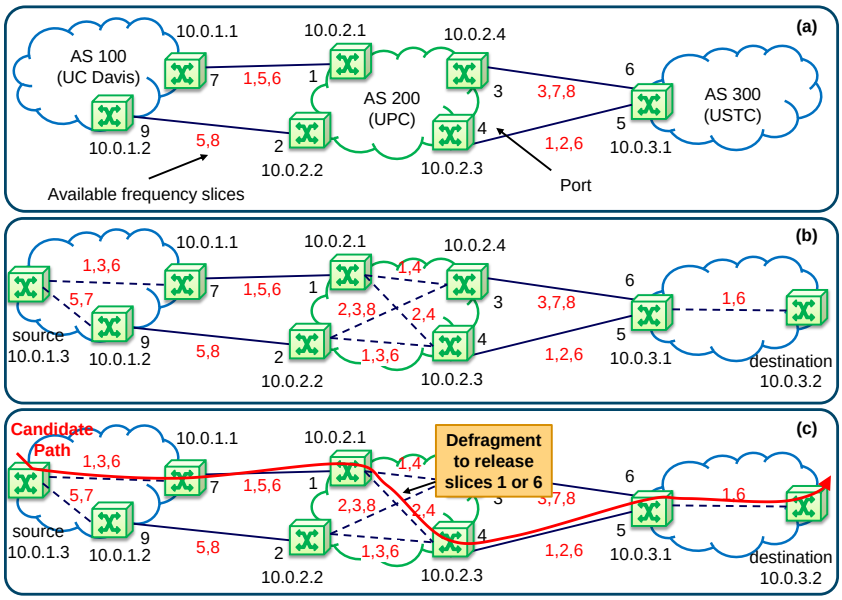

Figure 13: Example of path computation

Figure 14 illustrates the proposed provisioning workflow, which is divided into three main phases:

i) the *Domain Advertisement* phase is initiated when the broker first connects to the ASs controllers. The broker collects the inter-AS information, along with the AS's capabilities;

ii) the *Path Computation* phase is triggered by the arrival of a new inter-AS path computation request to an SDN controller. Next, the SDN controller forwards the request to the broker (step 5). Afterwards, the broker gets the intra-AS connectivity (steps 6 and 7). Then, the broker makes a path computation request to the planning tool, adding in the request message the new topology information just obtained (step 8). If the planning tool finds a feasible solution it responds to the broker the multi-AS

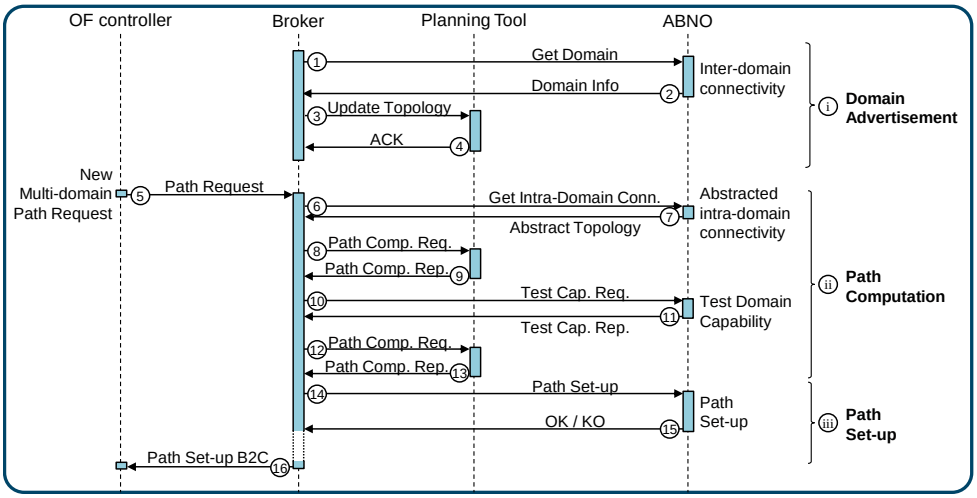

Figure 14: Proposed workflow

path to be set-up. Otherwise, it responds a no-path and proposes a solution using one or more capabilities (step 9). In the latter case, the broker tests if the capabilities are still available (steps 10 and 11). If the capabilities are successfully tested, the broker sends a new path computation request to the planning tool allowing the possibility of the using the just tested capabilities during the computation (step 12). Eventually, the planning tool responds with the multi-AS path to be set-upped and the list of capabilities to be used (step 13);

iii) in the *Path Set-up* phase, the broker, following the solution proposed by the planning tool, instructs the SDN controllers to signal the intra-AS path and configure the borders routers (steps 14 and 15). Once all the SDN controllers finish its local set-up, the broker informs the SDN controller which made the original request that the inter-AS path is signaled.

3.2.8.2 Experimental Assessment

The experimental validation was carried out on a distributed field trial set-up connecting premises in UC Davis (Davis, California), USTC (Hefei, China), and UPC (Barcelona, Spain) (Figure 12). The broker, the OF controllers and agents have been developed in Python and run in a computer cluster under Linux. The UPC's Planning tool for optical networks (PLATON) and the ABNO has been developed in C++ for Linux.

Regarding the management plane, to enable the broker to orchestrate the experiment, we have developed an HTTP REST API at the broker, which is implemented by the SDN controllers and PLATON. For each API function a specific XML has been devised. These XML files act as input/output parameters for the API functions (see Figure 15 and Figure 16).

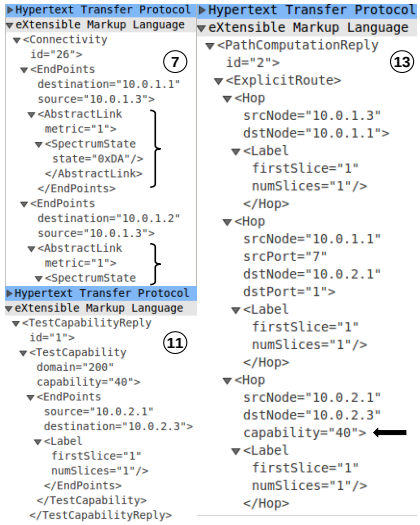

Figure 15: XML files for steps 7, 11, and 13

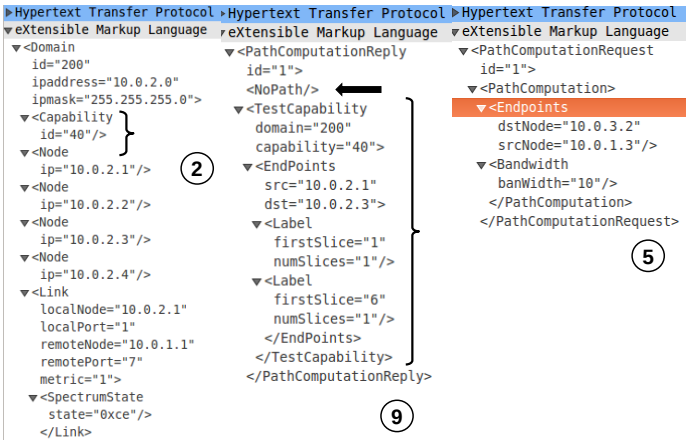

Figure 16: XML files for steps 2, 5, and 9

Figure 17 shows the exchanged messages from a broker point of view. For the sake of clarity, the numbers of the messages in the figures are in correspondence with each other.

No.	Time	Source	Destination	Protocol	Length	Info	
21502	18.732403000	169.237.74.210	222.195.92.93	HTTP/XML	① 323	GET /ctrl/GETDOMAIN HTTP/1.1	i
21505	18.920270000	222.195.92.93	169.237.74.210	HTTP/XML	② 617	HTTP/1.1 200 OK	
22907	19.146238000	169.237.74.210	147.83.42.198	HTTP/XML	③ 707	POST /platon/UPDATETOPOLOGY HTTP/1.1	
23740	19.375006000	147.83.42.198	169.237.74.210	HTTP/XML	④ 210	HTTP/1.0 200 OK	
24070	19.590218000	169.237.74.210	147.83.42.198	HTTP/XML	① 323	GET /ctrl/GETDOMAIN HTTP/1.1	
24584	19.816510000	147.83.42.198	169.237.74.210	HTTP/XML	② 870	HTTP/1.0 200 OK	
24601	20.033223000	169.237.74.210	147.83.42.198	HTTP/XML	③ 1044	POST /platon/UPDATETOPOLOGY HTTP/1.1	
25347	20.251706000	147.83.42.198	169.237.74.210	HTTP/XML	④ 210	HTTP/1.0 200 OK	
25487	20.258240000	169.237.74.210	169.237.74.208	HTTP/XML	① 324	GET /ctrl/GETDOMAIN HTTP/1.1	
25519	20.270053000	169.237.74.208	169.237.74.210	HTTP/XML	② 595	HTTP/1.1 200 OK	
25911	20.494557000	169.237.74.210	147.83.42.198	HTTP/XML	③ 685	POST /platon/UPDATETOPOLOGY HTTP/1.1	
25917	20.719023000	147.83.42.198	169.237.74.210	HTTP/XML	④ 210	HTTP/1.0 200 OK	
27573	35.731235000	169.237.74.208	169.237.74.210	HTTP/XML	⑤ 477	GET /ctrl/PathRequest HTTP/1.1	
27595	35.921073000	169.237.74.210	222.195.92.93	HTTP/XML	⑥ 456	GET /ctrl/GETINTRADOMCONN HTTP/1.1	ii
28071	36.109139000	222.195.92.93	169.237.74.210	HTTP/XML	⑦ 556	HTTP/1.1 200 OK	
28148	36.343476000	169.237.74.210	147.83.42.198	HTTP/XML	⑥ 564	GET /ctrl/GETINTRADOMCONN HTTP/1.1	
28157	36.585416000	147.83.42.198	169.237.74.210	HTTP/XML	⑦ 856	HTTP/1.0 200 OK	
28174	36.588595000	169.237.74.210	169.237.74.208	HTTP/XML	⑥ 402	GET /ctrl/GETINTRADOMCONN HTTP/1.1	
28185	36.591909000	169.237.74.208	169.237.74.210	HTTP/XML	⑦ 422	HTTP/1.1 200 OK	
28728	36.815523000	169.237.74.210	147.83.42.198	HTTP/XML	⑧ 1335	GET /platon/PCREQUEST HTTP/1.1	
28734	37.041866000	147.83.42.198	169.237.74.210	HTTP/XML	⑨ 449	HTTP/1.0 200 OK	
29251	37.270273000	169.237.74.210	147.83.42.198	HTTP/XML	⑩ 567	GET /ctrl/TCREQUEST HTTP/1.1	
29276	37.494772000	147.83.42.198	169.237.74.210	HTTP/XML	⑪ 404	HTTP/1.0 200 OK	
29300	37.712266000	169.237.74.210	147.83.42.198	HTTP/XML	⑫ 123	GET /platon/PCREQUEST HTTP/1.1	
29830	37.933499000	147.83.42.198	169.237.74.210	HTTP/XML	⑬ 810	HTTP/1.0 200 OK	
29855	38.149229000	169.237.74.210	147.83.42.198	HTTP/XML	⑭ 567	POST /ctrl/PATHSETUP HTTP/1.1	iii
29953	38.366745000	147.83.42.198	169.237.74.210	HTTP/XML	⑮ 209	HTTP/1.0 200 OK	
30398	38.554640000	169.237.74.210	222.195.92.93	HTTP/XML	⑭ 504	POST /ctrl/PATHSETUP HTTP/1.1	
30403	38.744328000	222.195.92.93	169.237.74.210	HTTP/XML	⑮ 232	HTTP/1.1 200 OK	
30412	38.746969000	169.237.74.210	169.237.74.208	HTTP/XML	⑭ 504	POST /ctrl/PATHSETUP HTTP/1.1	
30415	38.750651000	169.237.74.208	169.237.74.210	HTTP/XML	⑮ 232	HTTP/1.1 200 OK	
30423	38.753041000	169.237.74.210	169.237.74.208	HTTP/XML	⑯ 363	GET /ctrl/PATHSETUP_B2C HTTP/1.1	

Figure 17: Message Exchange at the broker

The workflow starts when the broker connects to all three SDN controllers and populates its topology. Every time a new topology is obtained, a copy is sent to PLATON, in order to maintain broker and PLATON databases synchronized (steps 1-4). In the event of a path computation request received from a SDN controller (step 5), the Broker collects abstracted intra-AS connectivity and AS capabilities from every controller (steps 6-7). Afterwards, the broker sends a path computation request to PLATON (step 8). In the path computation message, the broker also includes the new topology information just learned. PLATON, first updates its database with the new topology information contained in the request message, and then performs the path computation. Due to our set up, no solution is found. Consequently, a NoPath reply is sent to the broker. Within the reply message PLATON suggests that if defragmentation is used in the UPC AS, a solution can be found (step 9). Then, the broker accepts PLATON suggestion and tests the defragmentation capability in the UPC AS (step 10). As result of the test the UPC AS responds OK (step 11). Immediately after, the broker resends the path computation request to PLATON, but this time informing that the defragmentation capability can be used (step 12). Now PLATON finds a solution, and sends it to the broker. The solution in the path computation reply, the XML contains the routing and spectrum allocation, and the capability to be performed (step 13). Finally, the Broker creates the set of configurations to be forwarded to the corresponding SDN controllers (step 14). Eventually, when every controller confirms that the configuration has been set-up (step 15), the broker informs the requester SDN controller that the multi-AS path is signaled (step 16).

3.2.9 Experimental Demonstration of Heterogeneous Cross Stratum Broker for Scientific Applications

This year 3 activity is summarized in Table 16.

Table 16: Programmatic summary - experimental demonstration of heterogeneous cross stratum broker

Original Hypotheses	Orchestration for scientific applications and heterogeneous resources reservation in DCs, HPC facilities and networks belonging to different operators.
Approach Used	Propose and experimentally validate an architecture for scientific experiments to share computational facilities in geographically diverse locations and to provide a single entrance point to request heterogeneous connectivity.
Problems Encountered	None.
Departure from Methodology	None.
Impact on Project Results	A cross stratum heterogeneous Broker orchestrates resource reservation in DCs and HPC facilities and networks belonging to different operators. In particular, FPGAs available in DCs are used for data pre-processing and HPC facilities are used for computing complex scientific models.

Scientific applications often require intimate interactions between theoretical analysis and experimental measurements. Nowadays, scientific experiments demand increasingly more resources, such as storage and processing, challenging not only high performance computing (HPC), but also communications networks. In a scientific experiment, sensors detect events and generate data that is collected, pre-processed, and stored. Sensors can either all be placed in the same geographical experimental facility, like in the CERN's Large Hadron Collider and the IceCube Neutrino Observatory, or spread worldwide, such as in the Comprehensive Nuclear Test Ban Treaty Organization (CTBTO) sensor network. In such scientific experiments, sensors generate large amounts of data that contains both meaningful physical measurements and noise. Thus, data filtering and pre-processing is performed before even storing and transmitting data. Because of the significant data volume generated, dedicated hardware (Hw) (e.g., FPGAs) is frequently used. The final stage consists in running complex physical models, which requires a HPC facility. Although each experiment has its own needs, in general, they follow the aforementioned stages (Figure 18).

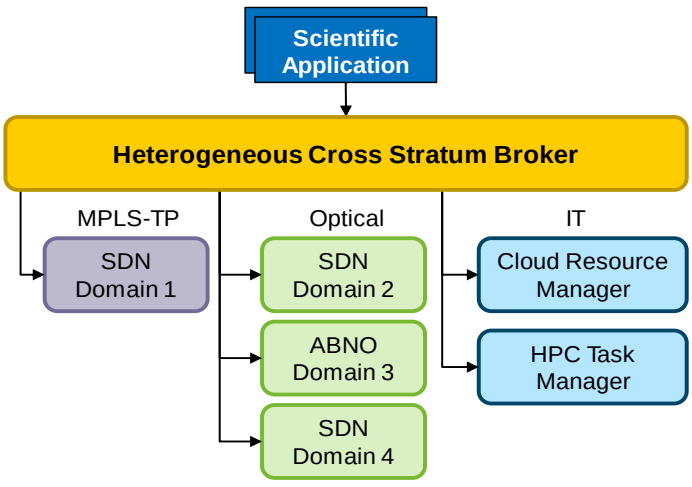

Figure 18: Broker as a single entrance point for scientific applications

The experimental facility and computational resources may belong to the same organization. However, since experiments are conducted from time to time, computational resources are underutilized, which entails a high cost. In view of that, we propose and experimentally validate an architecture for scientific experiments to share computational facilities in geographically diverse locations and to provide a single entrance point to request heterogeneous connectivity (i.e., MPLS-TP, WDM, and flexgrid) and IT (i.e., storage, specialized Hw, and HPC) resources.

3.2.9.1 Proposed architecture

Note that scientific experiments are usually carried out in two phases:

i) data collection and pre-processing, and
ii) model computation. Therefore, in between, pre-processed data must be stored and ready to be conveyed to the selected HPC facility. A scientific application must then be able to request specific resources to be provided immediately or to be reserved in advance. Another interesting requirement is the ability to reserve Hw resources (FPGA) located in datacenters (DC) to be loaded with the specific *bitstream* for the scientific experiment.

In consequence, we propose an architecture where scientific applications can request connectivity and IT resources to our Cross Stratum Broker, which is able to request IT resources (virtual machines, storage, and specialized Hw) to a set of DCs and computation slots at HPC facilities (Figure 19).

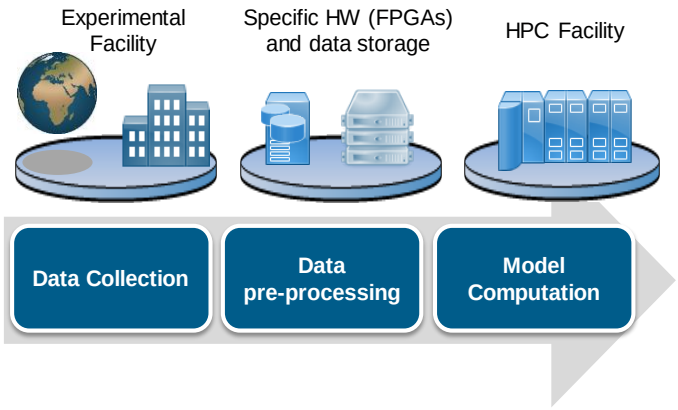

Figure 19: Extreme-scale scientific applications model

Since selected IT facilities could be placed in geographically distant locations and connected to different network operators, it is clear that the broker must support heterogeneous technologies at both data and control planes. To convey collected data to DCs either MPLS-TP or optical connections can be created, depending on the data volume. The same is applicable for conveying pre-processed data from DCs to the selected HPC facility. Figure 20 shows an example of distributed scenario connecting data collection, DC for data filtering and an HPC facility. Four domains are shown, where MPLS-TP domain 1 aggregates collected data towards optical domain 2 that transports collected data to the selected DC(s). Collected data is pre-processed in the FPGA using the specific bitstream for the scientific experiment and data is stored waiting to be sent to the HPC facility. Once all data is available and before the scheduled slot in the HPC facility, an end-to-end lightpath is set-up crossing three optical domains belonging to different network operators. Optical conversion capability at the different ingress and egress OCXs allow the broker to find a feasible end-to-end lightpath by performing conversion when no transparent spectrum can be found.

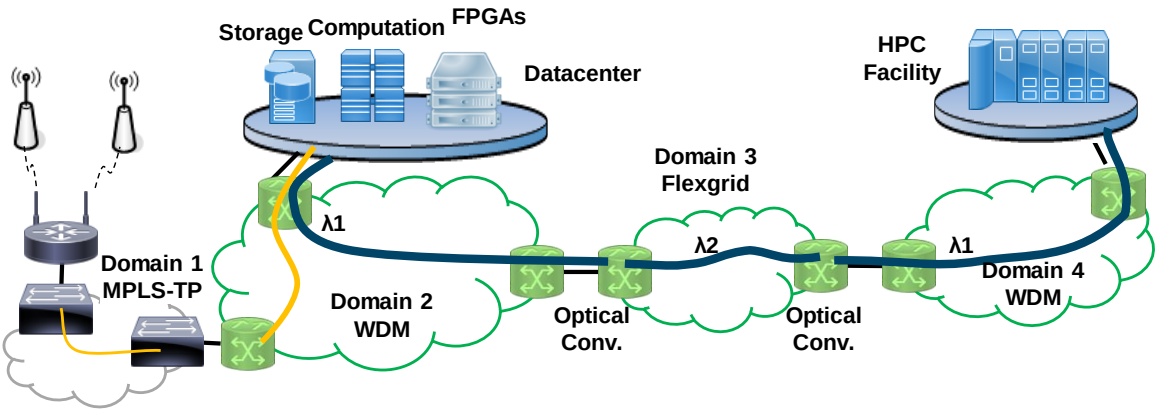

Figure 20: Example of distributed scenario connecting data collection, DC, and HPC

3.2.9.2 Scientific experiment set-up workflow

Before start accepting requests, the Broker needs to discover the available resources in every domain, i.e., storage, FPGAs, etc. in DCs, queues and priorities in HPCs, and network capabilities and inter-domain topology from each domain controller; a key network capability is optical conversion at inter-domain ingress and egress OXCs. When a scientific application needs resources for an experiment, it issues a request to the Broker specifying the IT and connectivity resources required together with some temporal constraints (step 1 in Figure 21). When the Broker receives such request, it collects the current status of the resources in every domain/facility (steps 2-3) and finds the set of resources that better fit the specific experiment needs.

In the case that some specific capability needs to be applied to release resources that are currently being used, the broker requests to apply such capability to the specific controller/manager. For instance, let us assume that the broker has found a path between the DC in domain 2 and the HPC facility in domain 4. However, no transparent wavelength assignment/spectrum allocation could be found. The broker might decide to apply the optical conversion capability in domain 3 to convert λ1 to λ2 (4-5). If a converter is available at the ingress and egress nodes in domain 3, the complete resource allocation can be performed. Therefore, the broker allocates resources in DCs (6-7), schedules jobs in HPC facilities (8-9), and establishes connections in the network domains (10-11). Finally, when the domain controllers confirm that all resources have been allocated, the Broker replies back to the scientific application informing the requested service availability and specifying the details of the job scheduling at the HPC facility to enable the scientific application to access the output results (12).

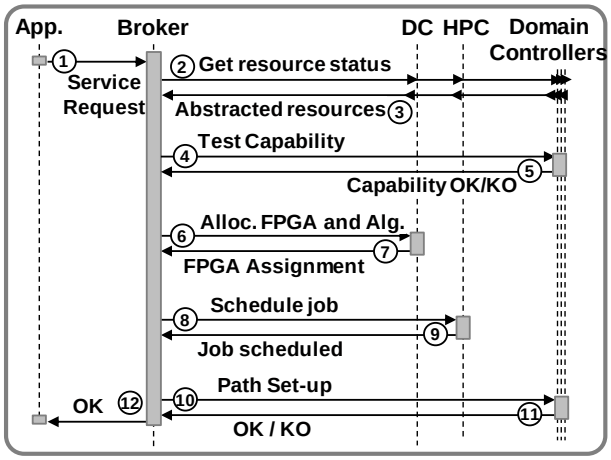

Figure 21: Proposed workflow

3.2.9.3 Experimental assessment

The experimental assessment has been carried out in a distributed test-bed spanning three continents. From the control plane, SDN/ABNO controllers have been deployed in Davis (USA) (domain 1 and 2), Barcelona (Spain) (domain 3), and Hefei (China) (domain 4). In contrast, the data plane, is in UCDavis labs, including data generation, OXCs with flexgrid WSSs and tunable lasers. Regarding the management plane, to enable the broker to orchestrate the experiment, we developed an HTTP REST API at the broker, which is implemented by the SDN controllers. For each API function a specific XML has been devised; these XML messages act as input/output parameters for the API functions. For the experiment, let us assume that an already established connection in domain 3 is using λ1, and the end-to-end connection from the DC in domain 2 to the HPC facility in domain 4 also needs λ1.

Figure 22 shows the exchanged messages from the broker viewpoint. For clarity purposes, message numbering used in the workflow has been also included. Despite not being shown in Figure 22 for space reasons, the workflow starts with the domains, DCs and HPC facilities advertisement [4]. Once the scientific application sends its request the broker triggers the proposed workflow.

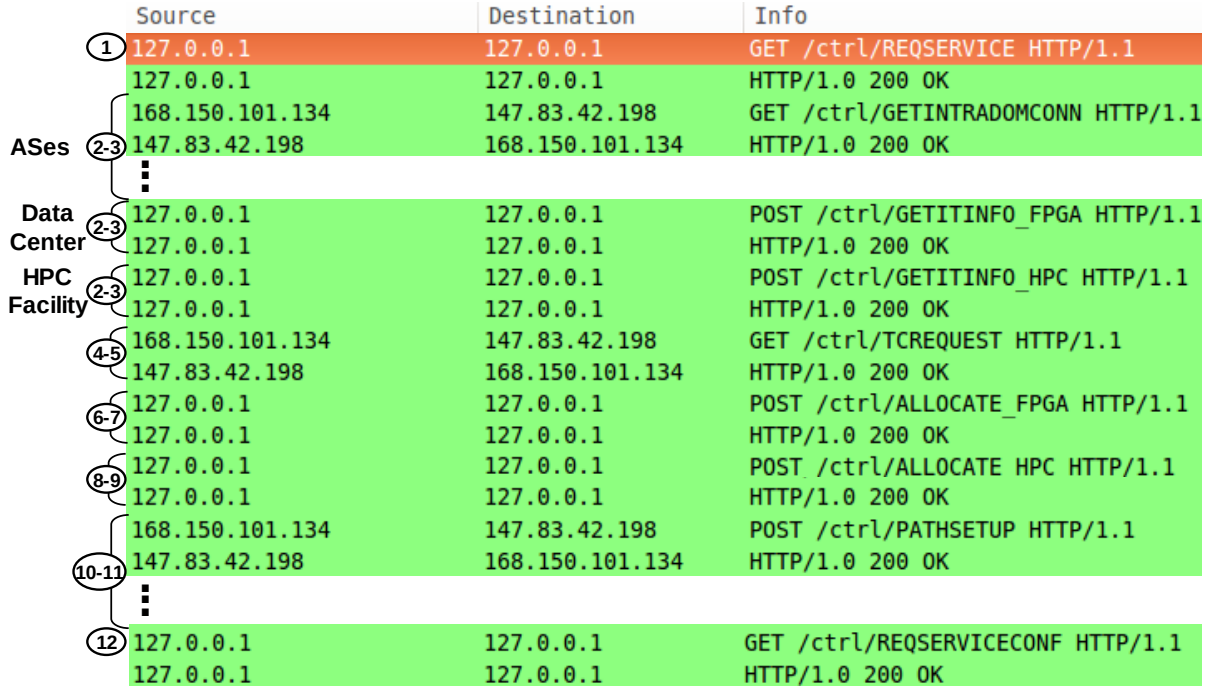

Figure 22: Message Exchange at the broker

Message 1 in Figure 23 depicts the service request received by the broker. The scientific application specifies the data source, which algorithms must be used to process the data, and the time constraint for the whole experiment. Note that, the scientific application can define as many constraints as needed.

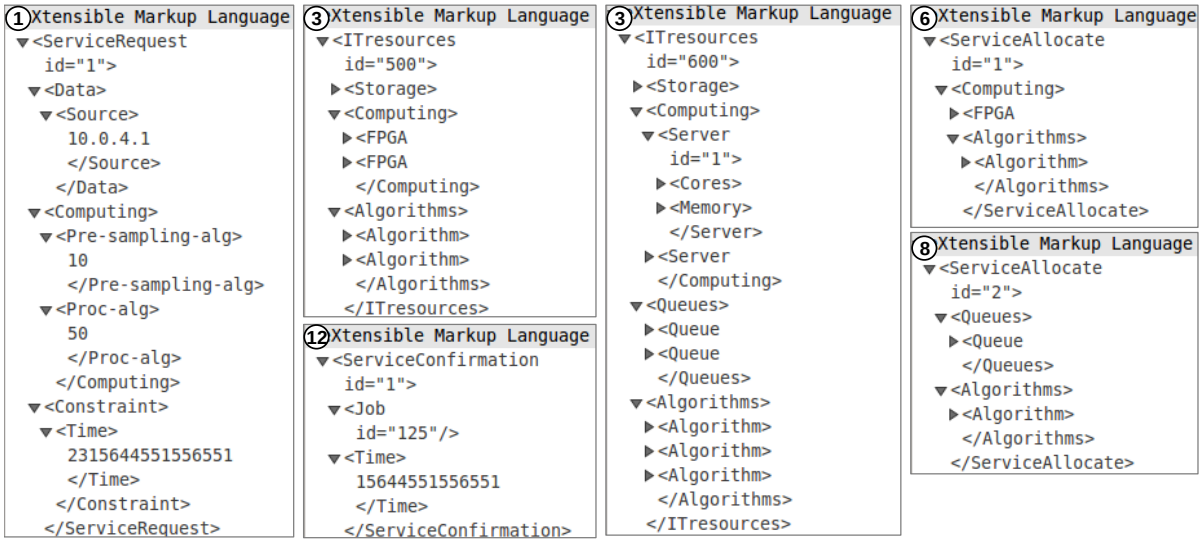

Figure 23: Detail of selected XML messages

Next, the broker updates its resources database (network and IT resources). Messages 3 in Figure 23 show the information exchange between the broker and the DC (left), and the HPC facility (right). In the case of DC, the broker gathers, e.g., the availability of the FPGAs and which algorithms can be instantiated. From the HPC facility data, the broker finds out the status of the computing queues, the available Hw resources and algorithms.

The workflow continues with the broker computing a solution, realizing that this solution implies applying a capability for conversion [4]. After the capability is confirmed, the broker allocates the IT resources. Messages 6 and 8 in **Error! Reference source not found.** show the messages sent by the broker to the DC and the HPC facility, respectively. Then, the path is set up. Eventually, the broker receives all the resource allocation confirmations, it replies to the scientific application with the *job id* for the request and the expected value of the constraints (see message 12 in Figure 23).

Figure 24a shows the node architecture for the ingress OXC in domain 3 (a similar architecture is used in the egress OXC). A flexgrid-enabled 1:4 WSS from Finisar implements the switching element in the OXCs. Output port 4 of the WSS implements the drop port connecting with the local interfaces module (client layer). A power coupler combines the signal from WSS's output port 2 with the signals coming from the local interfaces (add ports), which are multiplexed using another power coupler. We assume wavelength tunable local transponders. Figure 24b-d (left) show the optical spectrum at the OXC drop port (WSS port 4), whereas Figure 24b-d (right) show the spectrum at the add port (local interfaces coupler output A). In Figure 24b, WSS's port 4 is not configured and the already established connection occupying λ1 is shown. In **Error! Reference source not found.**c, the agent in the OXC has configured WSS's output 4 centered at λ1 and configured the O/E/O converter to use λ2. Finally, in Figure 24d, the optical connection is established and the λ1 signal received in port 4 is converted into λ2 and multiplexed with the already established connection using λ1.

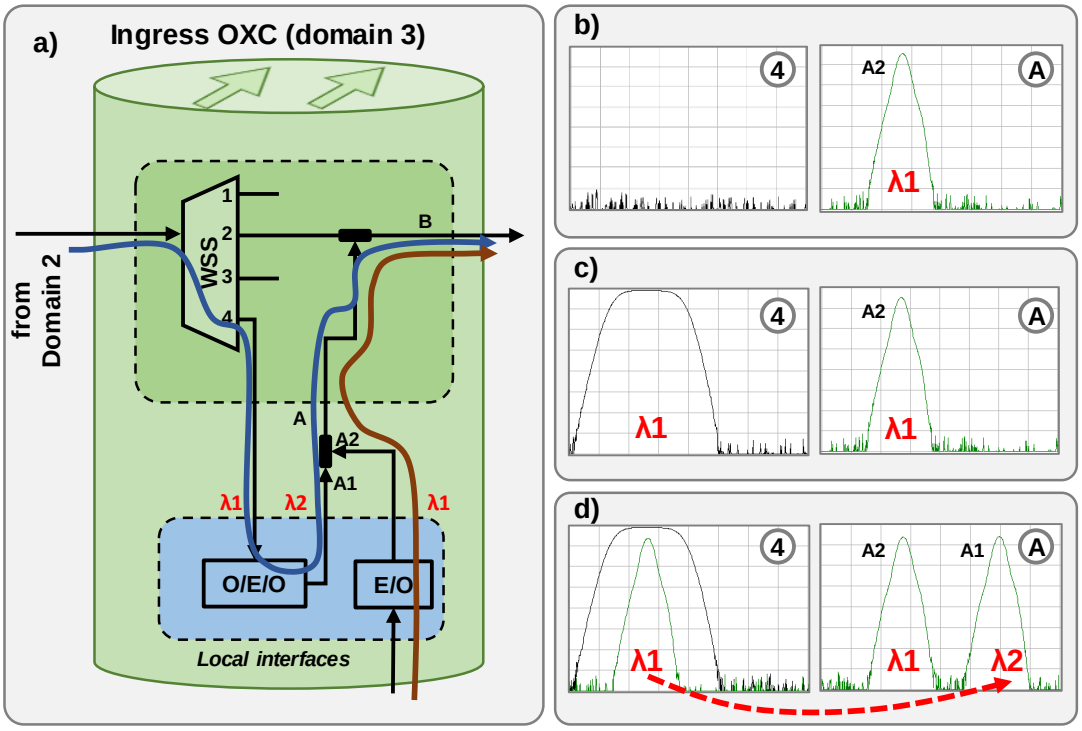

Figure 24: Experiments at the data plane

Figure 25 shows the experimental deployment.

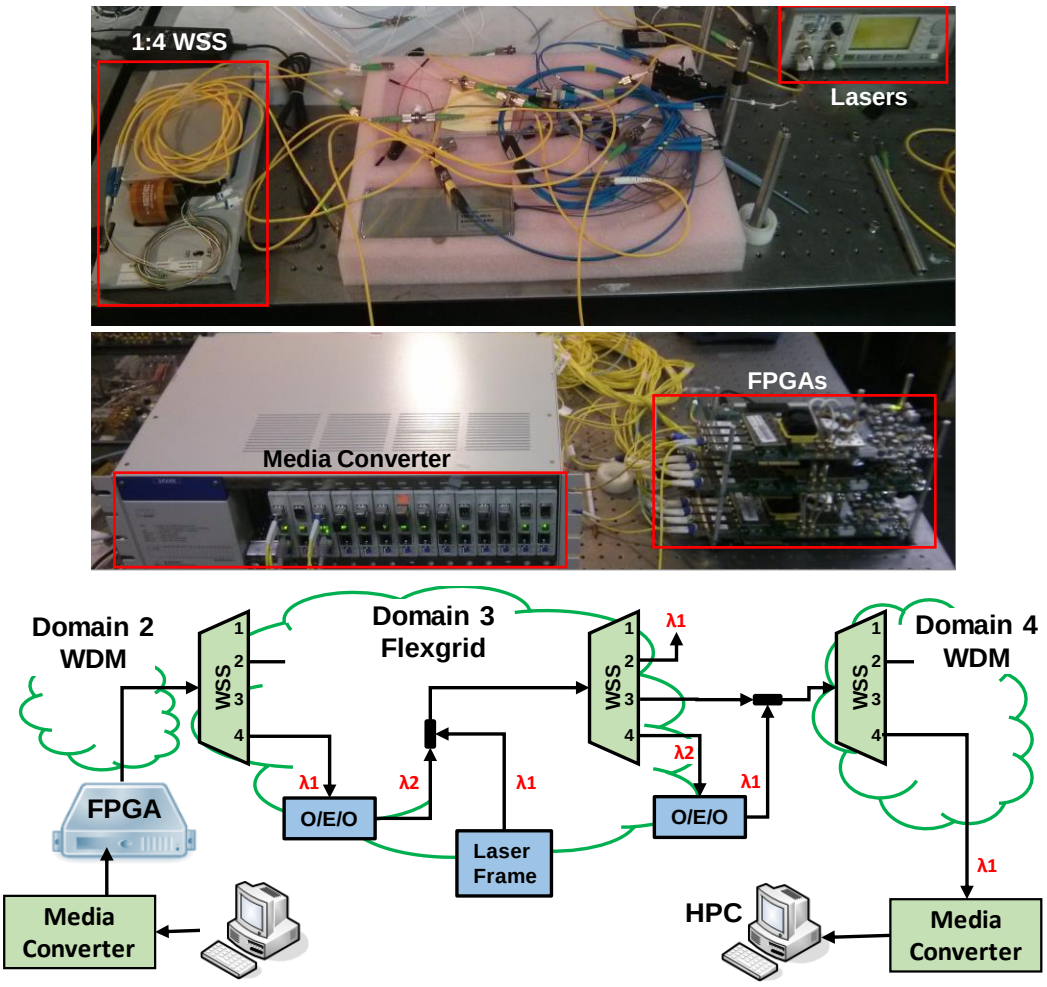

Figure 25: Experimental deployment

4 Publications, Products and Technology Transfers

Publications

[1] "Distributed control plane with spectral fragmentation-aware RMSA and flexible reservation for dynamic multi-domain software-defined elastic optical networks" by Hai-Chau Le, Lei Liu, and S. J. B. Yoo, accepted for publication in Optical Fiber Communication Conference (OFC), Paper Th2A.39, March, 2015.

[2] "Field trial of broker-based multi-domain software-defined heterogeneous wire-line-wireless-optical networks" by Lei Liu, Zuqing Zhu, Xiong Wang, Guanghua Song, Cen Chen, Xiaoliang Chen, Shoujiang Ma, Xiaotao Feng, Roberto Proietti, and S. J. B. Yoo, accepted for publication in Optical Fiber Communication Conference (OFC), Paper Th3J.5, March, 2015.

[3] "OpenFlow-Assisted Online Defragmentation in Single-/Multi-domain Software-Defined Elastic Optical Networks" by Zuqing Zhu, Xiaoliang Chen, Cen Chen, Shoujiang Ma, Mingyang Zhang, Lei Liu, and S. J. B. Yoo, in IEEE/OSA Journal of Optical Communications and Networking (Invited), Vol. 7, No. 1, pp. A7-A15, January, 2015.

[4] "Dynamic OpenFlow-based lightpath restoration in elastic optical networks on the GENI testbed" by Lei Liu, Wei-Ren Peng, Ramon Casellas, Takehiro Tsuritani, Itsuro Morita, Ricardo Martínez, Raül Muñoz, Masatoshi Suzuki, and S. J. B. Yoo , accepted for publication in IEEE/OSA Journal of Lightwave Technology (invited), 2015.

[5] "Demonstration of cooperative resource allocation in an OpenFlow-controlled multi-domain and multinational SD-EON testbed" by Zuqing Zhu, Cen Chen, Xiaoliang Chen, Shoujiang Ma, Lei Liu, Xiaotao Feng, and S. J. B. Yoo, accepted for publication in IEEE/OSA Journal of Lightwave Technology (invited), 2015.

[6] "3D Elastic Optical Networking in Temporal, Spectral, and Spatial Domains" by Roberto Proietti, Lei Liu, Ryan P. Scott, Binbin Guan, Chuan Qin, Tiehui Su, Francesco Giannone, and S. J. B. Yoo, accepted for publication in IEEE Communications Magazine, 2015.

[7] "Brokered Orchestration for End-to-End Service Provisioning across Heterogeneous Multi-Operator (Multi-AS) Optical Networks", Alberto Castro, Luis Velasco, Lluis Gifre, Cen Chen, Jie Yin, Zuqing Zhu, Roberto Proietti, and S. J. Ben Yoo, IEEE/OSA Journal of Lightwave Technology (JLT), 2016.

[8] "Incentive-Driven Bidding Strategy for Brokers to Compete for Service Provisioning Tasks in Multi-Domain SD-EONs", Xiaoliang Chen, Zuqing Zhu, Lu Sun, Jie Yin, Shilin Zhu, Alberto Castro, S. J. Ben Yoo, IEEE/OSA Journal of Lightwave Technology (JLT), 2016.

[9] "On-Demand Incremental Capacity Planning in Optical Transport Networks", Luis Velasco, Fernando Morales, Lluís Gifre, Alberto Castro, Oscar González de Dios, and Marc Ruiz, IEEE/OSA Journal of Optical Communications and Networking (JOCN), 2016.

[10] "Elastic Optical Networking by Dynamic Optical Arbitrary Waveform Generation and Measurement", Roberto Proietti, Chuan Qin, Binbin Guan, Nicolas K. Fontaine,

Shaoqi Feng, Alberto Castro, S. J. Ben Yoo, IEEE/OSA Journal of Optical Communications and Networking (JOCN), 2016.

[11] "Broker-based Multi-Task Gaming to Facilitate Profit-Driven Network Orchestration in Multi-Domain SD-EONs", Lu Sun, Shilin Zhu, Xiaoliang Chen, Zuqing Zhu, Alberto Castro, and S.J.B. Yoo, Optical Fiber Communication Conference (OFC), 2016.

[12] "Experimental Demonstration of Heterogeneous Cross Stratum Broker for Scientific Applications", Alberto Castro; Alba Perez; Lluis Gifre; Roberto Proietti; Chen Cen; Yie Jie; Xiaoliang Chen; Zheng Cao; Zuqing Zhu; Vinod Mishra; Luis Velasco; S. J. B. Yoo, Optical Fiber Communication Conference (OFC), 2016.

[13] "Experimental Demonstration of Brokered Orchestration for end-to-end Service Provisioning and Interoperability across Heterogeneous Multi-Operator (Multi-AS) Optical Networks", Alberto Castro, Lluis Gifre, Cen Chen, Jie Yin, Zuqing Zhu, Luis Velasco, S. J. Ben Yoo, European Conference on Optical Communications (ECOC), 2015.

[14] "Multi-Broker based Market-Driven Service Provisioning in Multi-Domain SD-EONs in Noncooperative Game Scenarios", Xiaoliang Chen, Jie Yin, Cen Chen, Zuqing Zhu, Alberto Castro, S. J. Ben Yoo, European Conference on Optical Communications (ECOC), 2015.

Software

The DRC software is distributed based on a BSD-3 open-source license, which allows for very liberal duplication and inclusion in commercial products. The code for DRC is available upon request.